LA-UR-20-23516
Approved for public release; distribution is unlimited.

Title: Hamiltonian Simulation

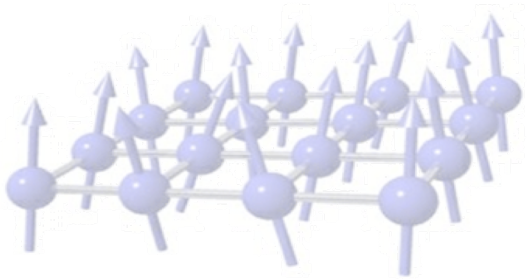
Author(s): Somma, Rolando Diego

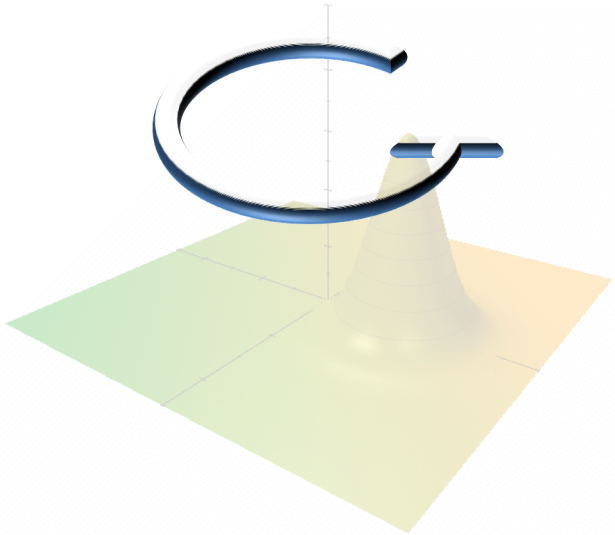
Intended for: Seminar

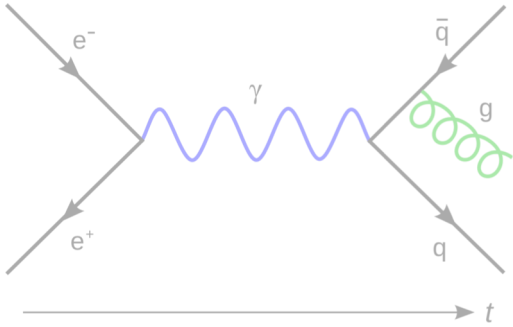
Issued: 2020-05-11

Hamiltonian Simulation

Rolando D. Somma
Theoretical Division
Los Alamos National Laboratory







University of Washington
May 12, 2020

Los Alamos NATIONAL LABORATORY EST. 1943

The Hamiltonian Simulation Problem in a Nutshell

- Given a (qubit) quantum system modeled by a Hamiltonian *H*, use a quantum computer to obtain time-evolved properties of the system.

- One way of solving this problem is by actually preparing the evolved quantum state (in some encoded fashion).

- The goal is to provide a sequence of elementary 1 and 2-qubit gates that would prepare such a state.

The Hamiltonian Simulation Problem in a Nutshell

- Given a (qubit) quantum system modeled by a Hamiltonian *H*, use a quantum computer to obtain time-evolved properties of the system.

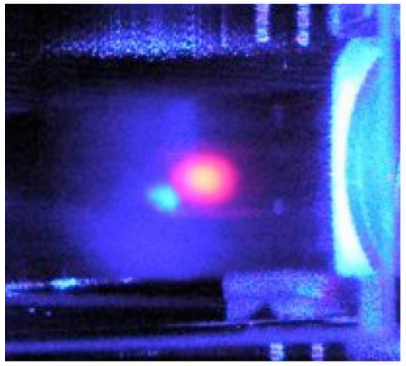
- One way of solving this problem is by actually preparing the evolved quantum state (in some encoded fashion).

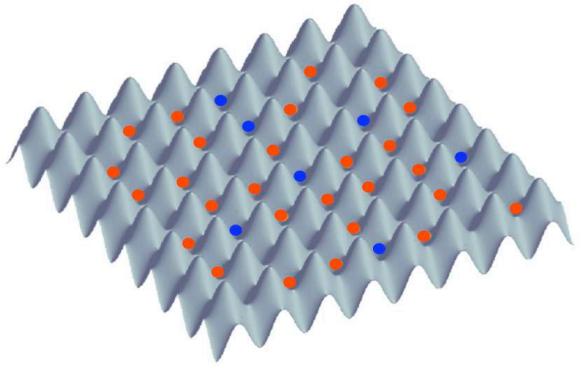
- The goal is to provide a sequence of elementary 1 and 2-qubit gates that would prepare such a state.

My talk is about this

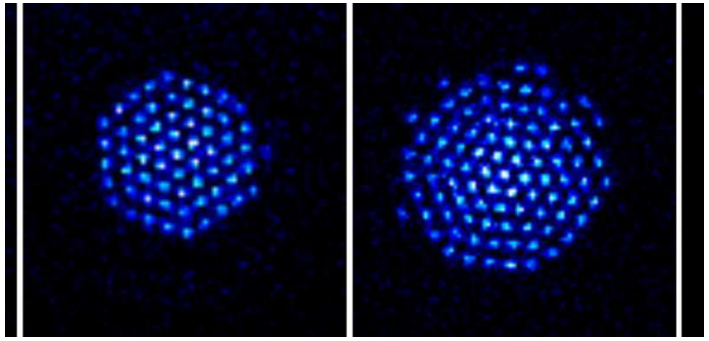
Quantum Simulators as Special-Purpose Quantum Computers

Optical lattices



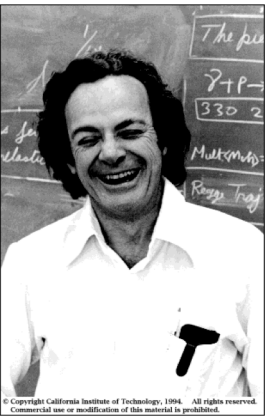


Ion traps



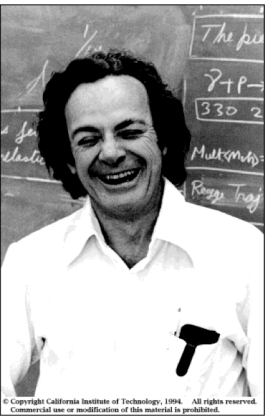
- The idea is to design the interactions such that they mimic the Hamiltonian *H*
- Such interactions are always ‘ON’ or their control is limited - there are no gates
- **I will not talk about this!** I will talk about universal quantum-computer simulations

Quantum Systems and Quantum Computers

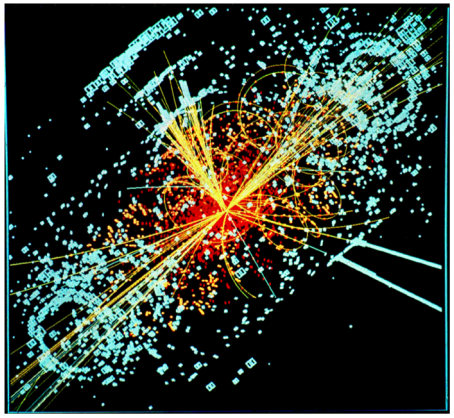


- The simulation of quantum systems was the main motivation behind Feynman's proposal of a quantum computer.
- For example, simulating n-spin systems with conventional computers require time exponential in n. Quantum computers are, in principle, not subject to this "exponential explosion".

Quantum Systems and Quantum Computers



- The simulation of quantum systems was the main motivation behind Feynman's proposal of a quantum computer.
- For example, simulating n-spin systems with conventional computers require time exponential in n. Quantum computers are, in principle, not subject to this "exponential explosion".



Numerous applications:

- Computation of scattering amplitudes, correlation functions, equilibrium properties, …
- Quantum chemistry: simulation of new chemical reactions; Quantum field theories; Quantum transport;…
- Quantum simulation appears as a subroutine in many algorithms (Linear systems, quantum simulated annealing,…)

Quantum Systems and Quantum Computers

Moreover:

- Many problems related to computing properties of quantum systems are BQP-hard or BQP-Complete [Wocjan and Zhang, 2006]
- In this sense, Hamiltonian Simulation is the PROBLEM for quantum computers

SCIENTIFIC AMERICAN CELEBRATING 175 YEARS

Are We Ready for Quantum Computers?

Hardware hasn't caught up with theory, but we're already lining up many previously intractable problems for when it does

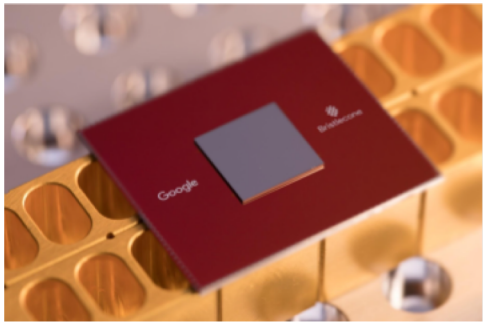
By Rolando Somma on March 13, 2020

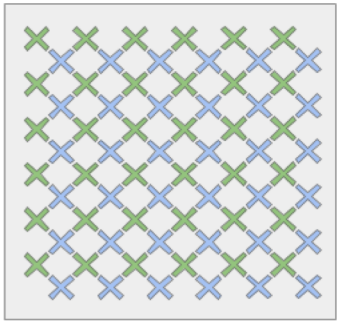
…. Simulating quantum physics is *the* app for quantum computers. They're not going to be helping you stream video on your smartphone. If ….

Near Term Quantum Computers

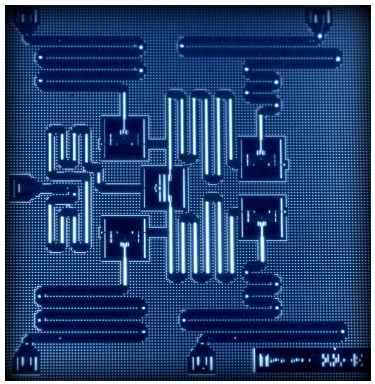
Something related to the simulation of physical systems *may* be the main **application** of near-term quantum computers:

- Order of 100s qubits
- Limits on # of gates

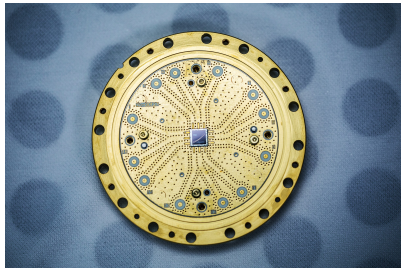




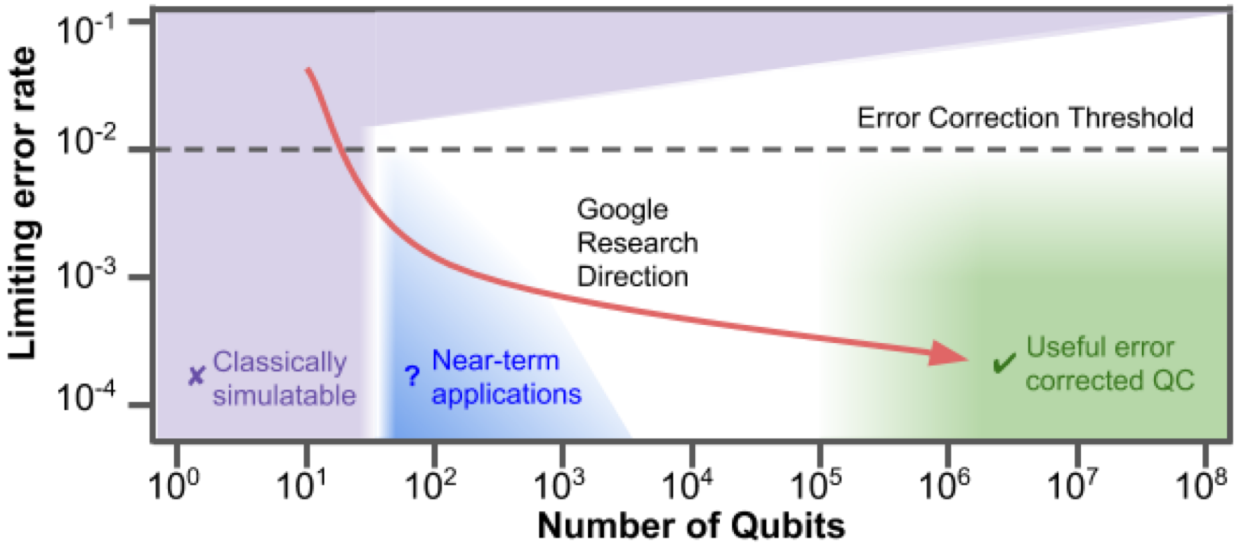
Google Bristlecone



IBM



Rigetti



Near Term Quantum Computers

A big issue: quantum error correction is left under the rug!

Can we really simulate physical systems with these devices?
How big? What problems? Dynamics, equilibrium?

Many open questions. A lot of interest in this recently: any useful contribution is extremely important, and it is often the case that Hamiltonian simulation methods provide the basis of near-term methods.

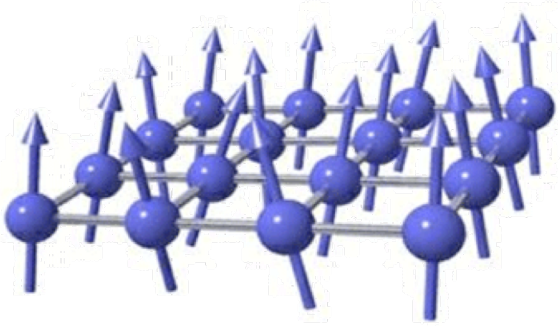
Quantum Simulation: Motivations, Outline

- Quantum simulation is an extremely active field of research
- Nearly every day there is at least one arXiv paper posted on this topic, if not more
- While a lot of progress has been made, many important and open questions remain. In fact, improved Hamiltonian simulation methods will be needed for a first demonstration.

I will summarize some of the most useful techniques and results, but there won't be enough time to go over every result

Also, I will pick simple examples, but the results I'll present can be generalized to more complex quantum systems and could address other models, such as the quantum query model

Hamiltonian Simulation: A Simple Example

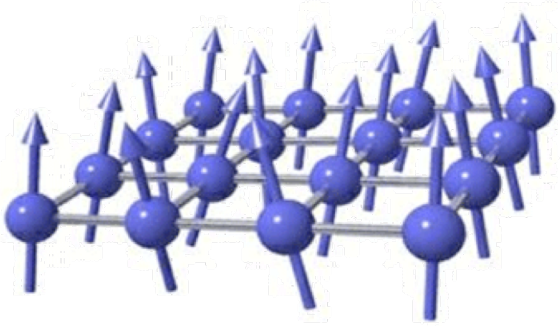


n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

Hamiltonian Simulation: A Simple Example



n-qubit (spin-1/2) system

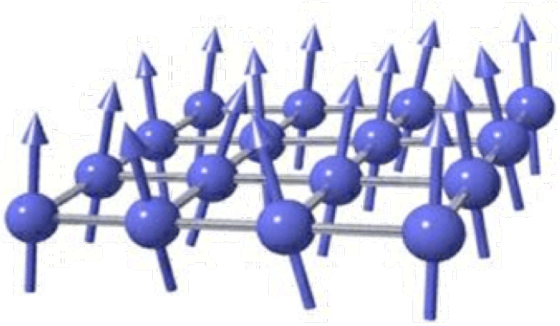
Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

products of Pauli operators

real numbers (0, 1]

Hamiltonian Simulation: A Simple Example



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

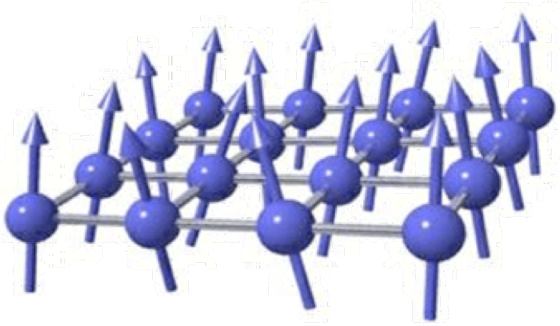
$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

products of Pauli operators

real numbers (0, 1]

For example $\hat{O}_k = X^j X^{j+1}$

Hamiltonian Simulation: A Simple Example



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

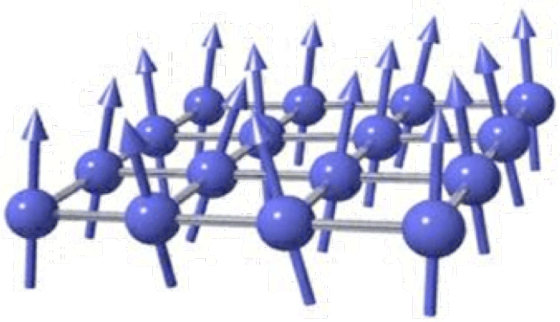
products of Pauli operators

real numbers (0, 1]

For example $\hat{O}_k = X^j X^{j+1}$

$$X^j = I \otimes \ldots \otimes I \otimes \underbrace{X}_{j\text{-th position}} \otimes I \ldots \otimes I$$

Hamiltonian Simulation: A Simple Example



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

products of Pauli operators

real numbers (0, 1]

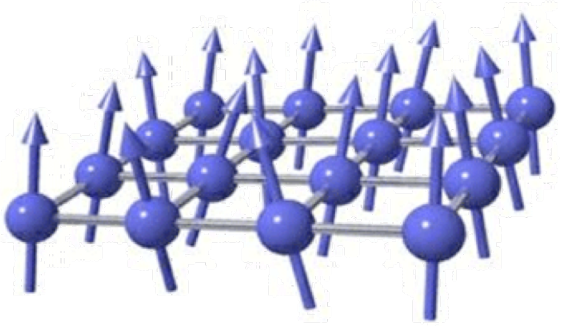
For example $\hat{O}_k = X^j X^{j+1}$

$$X^j = I \otimes \ldots \otimes I \otimes \underbrace{X}_{j\text{-th position}} \otimes I \ldots \otimes I$$

Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Hamiltonian Simulation: A Simple Example



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

products of Pauli operators

real numbers (0, 1]

For example $\hat{O}_k = X^j X^{j+1}$

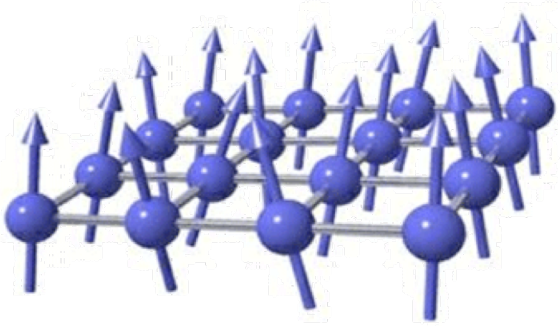
$$X^j = I \otimes \ldots \otimes I \otimes \underbrace{X}_{j\text{-th position}} \otimes I \ldots \otimes I$$

In matrix representation, these are of dimension NxN, where $N=2^n$

Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Hamiltonian Simulation: Time Evolution



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

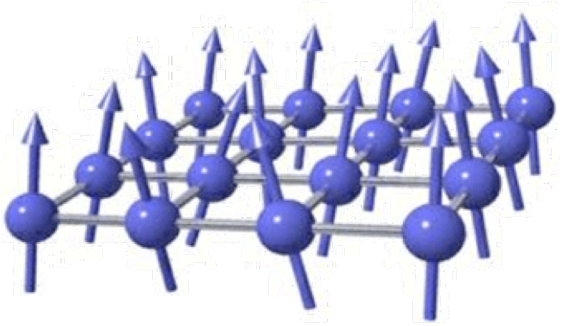
Hamiltonian simulation: **Approximating time evolution**

Given the Hamiltonian *H* (say as a l.c. of Paulis), an evolution time t>0, and a precision parameter ε>0, construct a quantum circuit $V=V_{L-1}\ldots V_1 V_0$ such that

$$\| e^{-iHt} |\psi\rangle - V|\psi\rangle \| \leq \epsilon$$

for all quantum states $|\psi\rangle$

Hamiltonian Simulation: Time Evolution



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

Hamiltonian simulation: **Approximating time evolution**

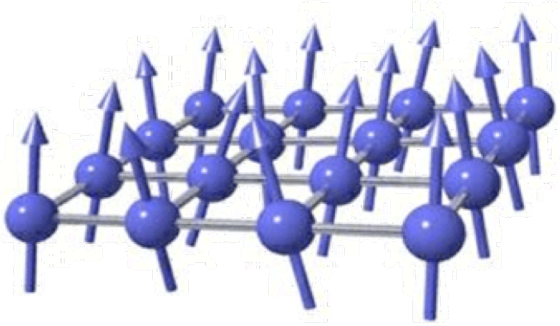
Given the Hamiltonian *H* (say as a l.c. of Paulis), an evolution time t>0, and a precision parameter ε>0, construct a quantum circuit $V=V_{L-1}\ldots V_1 V_0$ such that

$$\| |0\ldots 0\rangle (e^{-iHt}|\psi\rangle) - V|0\ldots 0\rangle|\psi\rangle \| \leq \epsilon$$

for all quantum states $|\psi\rangle$

The main objective is to develop a quantum algorithm that prepares the state and has gate complexity that is polynomial in the system size n and other relevant parameters. We will call this algorithm efficient.

Quantum Simulations: Classical Complexity



n-qubit (spin-1/2) system

Modeled by a Hamiltonian

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

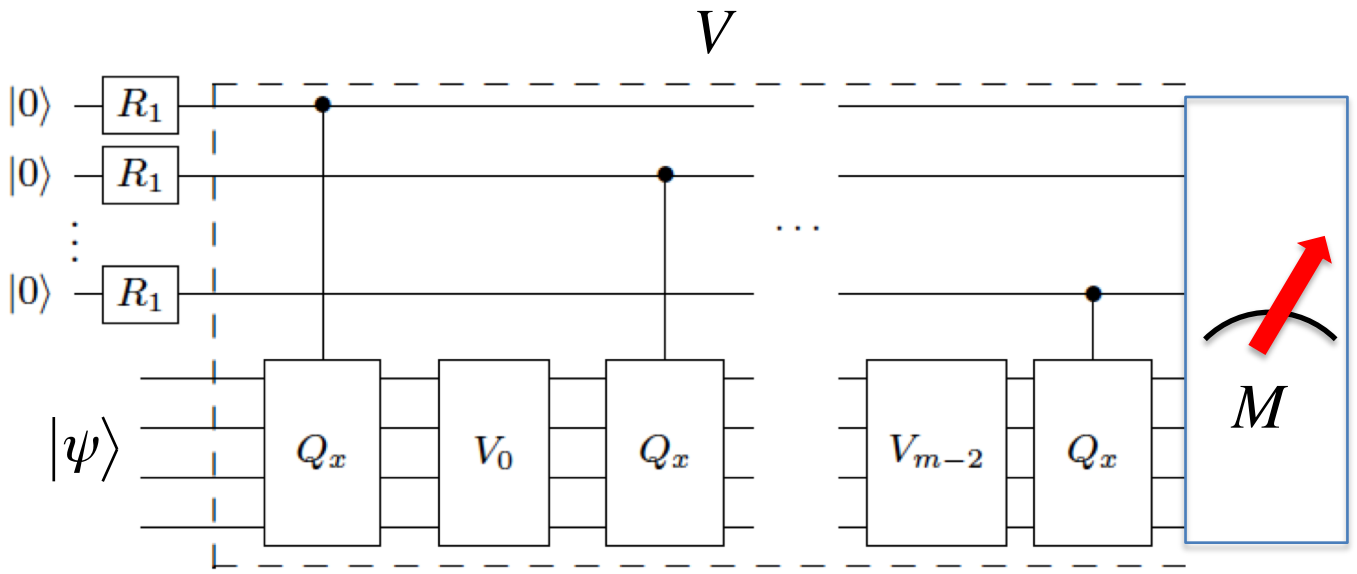
Note: classical algorithms for computing properties of an n-qubit system require, in the worst case, dealing with matrices of exponential dimension in n. Therefore, the quantum simulation problem is one of those problems that may be solved exponentially faster on a quantum computer.

Quantum Algorithms and Hamiltonian Simulation

1- Preparation of an initial pure state $|0\ldots0\rangle|\psi\rangle$

2- Unitary evolution $V=V_{L-1}\ldots V_1V_0$

3- Measurement of observable M

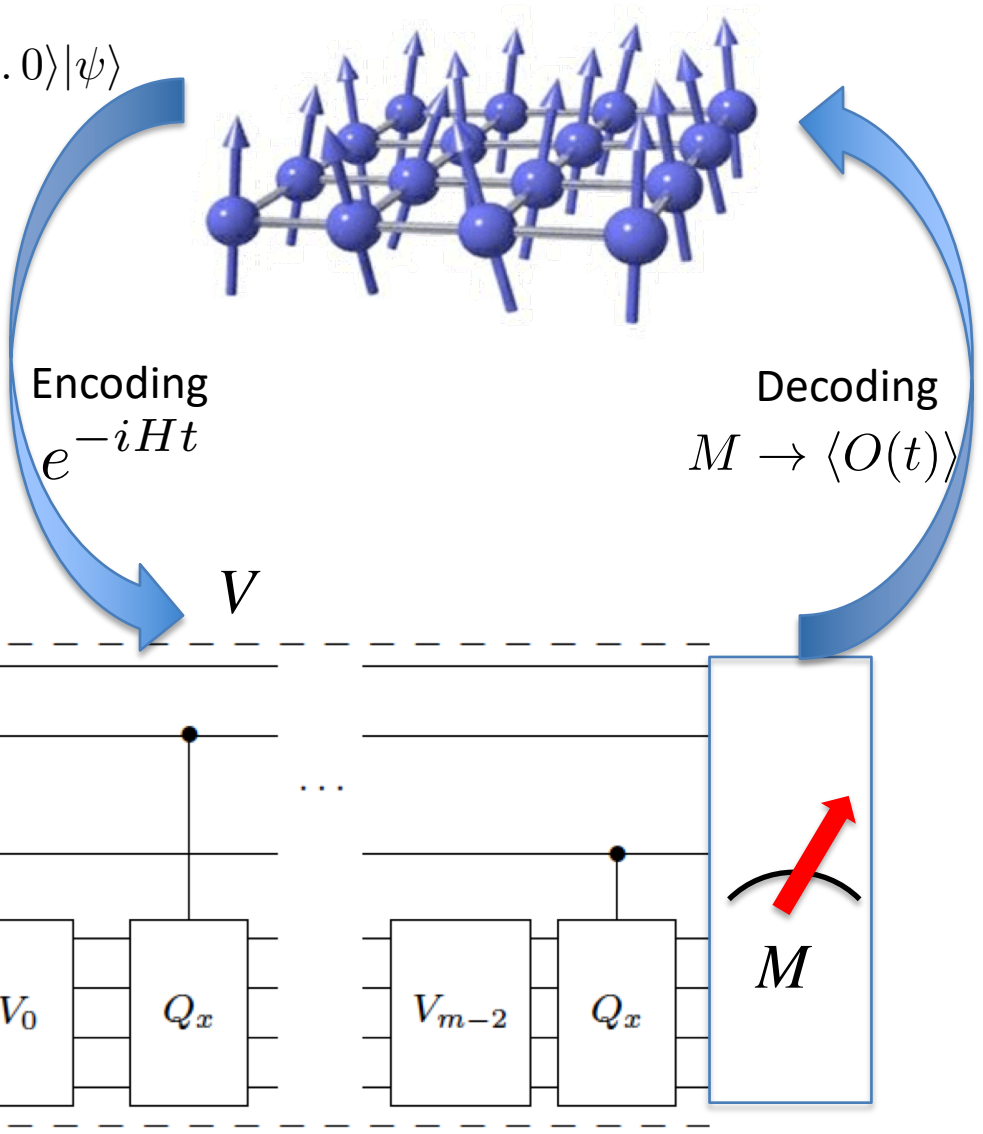


Quantum Algorithms and Hamiltonian Simulation

1- Preparation of an initial pure state $|0\ldots0\rangle|\psi\rangle$

2- Unitary evolution $V=V_{L-1}\ldots V_1V_0$

3- Measurement of observable M

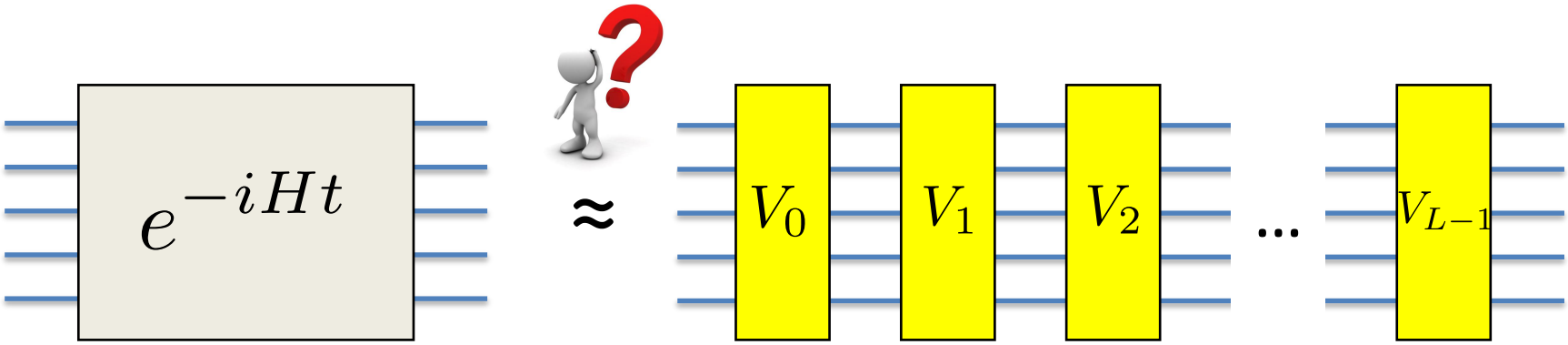


Quantum Algorithms and Hamiltonian simulation

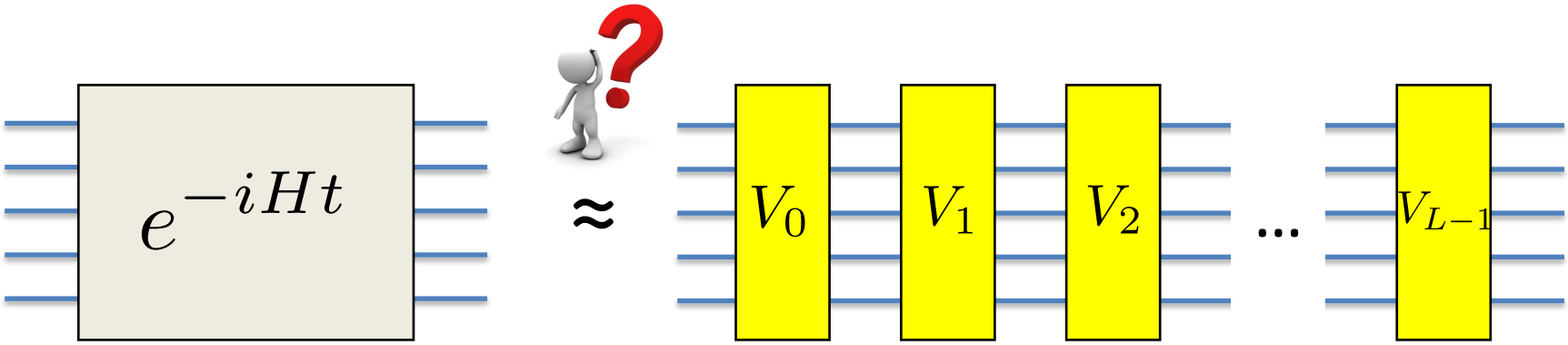
When constructing quantum algorithms, we need to focus on resources:

- Total number of qubits
- Number of fresh ancillary qubits
- Number of measurements
- Number of 1- and 2-qubit elementary gates

Quantum Computing and Hamiltonian Simulation



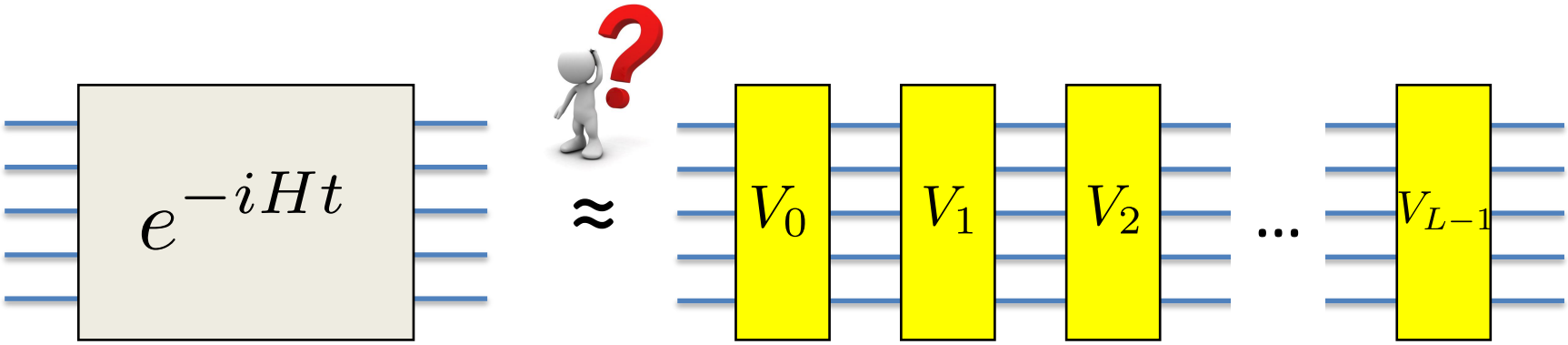
Quantum Computing and Hamiltonian Simulation



How can we approximate the exponential operator by a sequence of elementary gates?

We want to achieve the goal not only by providing an efficient quantum algorithm, but also by providing the best possible method in terms of overall resources.

Quantum Computing and Hamiltonian Simulation



In the following, I will revisit two of the most successful methods to approximate time evolution on a quantum computer.

1) Trotter-Suzuki approximations

2) Taylor-series approximation (linear combination of unitaries, LCU)

Hamiltonian Simulation: Trotter-Suzuki Approximations

- These (high-order) approximations are attractive due to their simplicity, they are intuitive, and do not need of ancillary qubits, making them attractive for near term applications.

- They basically approximate the evolution operator by short time evolutions under its constituents, each being simulated using a few 1 and 2-qubit gates.

- An upside is that they exploit some structure of the problem (e.g., commutation relations) and in such cases they perform very well. But if such structure can not be exploited or is unknown, the performance is poor.

- Another downside is that their complexity in terms of accuracy is also poor and can be prohibitive for some applications.

Hamiltonian Simulation: Trotter-Suzuki Approximations

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(t) = e^{-iHt}$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(t) = e^{-iHt}$$

$$U(t) \approx \ldots e^{-is_K \hat{O}_0} . e^{-is_{K-1} \hat{O}_{K-1}} \ldots e^{-is_1 \hat{O}_1} . e^{-is_0 \hat{O}_0}$$

$$s_j \ll t$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(t) = e^{-iHt}$$

$$U(t) \approx \ldots e^{-is_K \hat{O}_0} . e^{-is_{K-1} \hat{O}_{K-1}} \ldots e^{-is_1 \hat{O}_1} . e^{-is_0 \hat{O}_0}$$

$$s_j \ll t$$

- The time evolution of H is approximated by a sequence of short-time evolutions under each operator in the Hamiltonian.
- The approximation becomes an equality in the limit of s_j=0.
- Each short-time evolution corresponds to either an elementary gate or can be easily decomposed as a sequence of elementary gates following standard methods.
- The complexity of the quantum algorithm is mainly determined by the number of short-time evolutions in the product. **Note that no ancillary qubits are necessary!**

Hamiltonian Simulation: Trotter-Suzuki Approximations

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(t) = e^{-iHt}$$

$$U(t) \approx \ldots e^{-is_K \hat{O}_0} . e^{-is_{K-1} \hat{O}_{K-1}} \ldots e^{-is_1 \hat{O}_1} . e^{-is_0 \hat{O}_0}$$

$$s_j \ll t$$

- The time evolution of H is approximated by a sequence of short-time evolutions under each operator in the Hamiltonian.
- The approximation becomes an equality in the limit of s_j=0.
- Each short-time evolution corresponds to either an elementary gate or can be easily decomposed as a sequence of elementary gates following standard methods.

$$e^{-isX^1Y^2Y^5} = e^{i\pi Z^2Y^5/4} . e^{-isX^2Z^2} . e^{-i\pi Z^2Y^5/4}$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

- There is a recipe to choose the evolution times appearing in the product

The simplest case: first order approximation

$$H = \hat{O}_0 + \hat{O}_1 \Rightarrow W_1(s) = e^{-is\hat{O}_0} e^{-is\hat{O}_1} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

- There is a recipe to choose the evolution times appearing in the product

The simplest case: first order approximation

$$H = \hat{O}_0 + \hat{O}_1 \Rightarrow W_1(s) = e^{-is\hat{O}_0} e^{-is\hat{O}_1} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$U(s) = e^{-isH} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$\boxed{\|U(s) - W_1(s)\| = \mathcal{O}(s^2)}$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

- There is a recipe to choose the evolution times appearing in the product

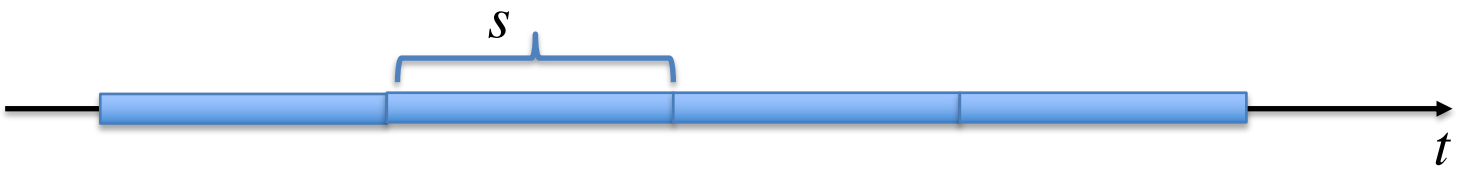
The simplest case: first order approximation

$$H = \hat{O}_0 + \hat{O}_1 \Rightarrow W_1(s) = e^{-is\hat{O}_0} e^{-is\hat{O}_1} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$U(s) = e^{-isH} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$\|U(s) - W_1(s)\| = \mathcal{O}(s^2)$$

Split the evolution into segments:



$$r = t/s \qquad U(t) \approx_\epsilon (W_1(s))^r$$

Trotter number

Hamiltonian Simulation: Trotter-Suzuki Approximations

- There is a recipe to choose the evolution times appearing in the product

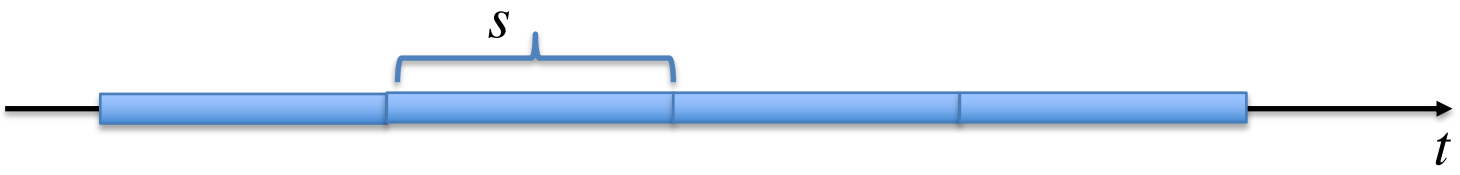
The simplest case: first order approximation

$$H = \hat{O}_0 + \hat{O}_1 \Rightarrow W_1(s) = e^{-is\hat{O}_0} e^{-is\hat{O}_1} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$U(s) = e^{-isH} = I - is(\hat{O}_0 + \hat{O}_1) + \mathcal{O}(s^2)$$

$$\|U(s) - W_1(s)\| = \mathcal{O}(s^2)$$

Split the evolution into segments:



$r = t/s$ Trotter number

$$U(t) \approx_\epsilon (W_1(s))^r$$

$$rs^2 \leq \epsilon \rightarrow r\ t^2/\epsilon$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

Complexity in high-order Trotter-Suzuki approximations

Recursive relation

$$W_1(s) = \prod_{k=0}^{K-1} e^{-isc_k\hat{O}_k}$$

$$W_2(s) = \prod_{k=0}^{K-1} e^{-isc_k\hat{O}_k/2} \prod_{k=K-1}^{0} e^{-isc_k\hat{O}_k/2}$$

$$W_{2p}(s) = [W_{2p-2}(r_k s)]^2 \, W_{2p-2}((1-4r_k)s) \, [W_{2p-2}(r_k s)]^2$$

$$r_k = 1/(4 - 4^{1/(2k-1)})$$

$$\|U(s) - W_p(s)\| = \mathcal{O}((Ks)^{p+1})$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

Complexity in high-order Trotter-Suzuki approximations

Recursive relation

$$W_1(s) = \prod_{k=0}^{K-1} e^{-isc_k\hat{O}_k}$$

$$W_2(s) = \prod_{k=0}^{K-1} e^{-isc_k\hat{O}_k/2} \prod_{k=K-1}^{0} e^{-isc_k\hat{O}_k/2}$$

$$W_{2p}(s) = [W_{2p-2}(r_k s)]^2 \, W_{2p-2}((1-4r_k)s) \, [W_{2p-2}(r_k s)]^2$$

$$r_k = 1/(4 - 4^{1/(2k-1)})$$

$$\|U(s) - W_p(s)\| = \mathcal{O}((Ks)^{p+1})$$

$$U(t) \approx_\epsilon (W_p(s))^r$$

$$r \sim \frac{(Kt)^{1+1/p}}{\epsilon^{1/p}}$$

$$G \sim 5^p K r$$

G: number of terms in product, “gate complexity”

Hamiltonian Simulation: Trotter-Suzuki Approximations

- Trotter-Suzuki approximations of the evolution operator provide then an efficient quantum algorithm: No complexity is polynomial in the dimension N, only linear in the number of terms in the Hamiltonian K, which scales with the system size n

$$G \sim 5^p K \frac{(Kt)^{1+1/p}}{\epsilon^{1/p}}$$

Hamiltonian Simulation: Trotter-Suzuki Approximations

- Trotter-Suzuki approximations of the evolution operator provide then an efficient quantum algorithm: No complexity is polynomial in the dimension N, only linear in the number of terms in the Hamiltonian K, which scales with the system size n

$$G \sim 5^p K \frac{(Kt)^{1+1/p}}{\epsilon^{1/p}}$$

- No ancillary qubits!

- The method is efficient, but how well does it perform in practice? Polynomial scaling may not be enough if a large number of elementary gates still needs to be implemented to show an advantage with respect to classical-computer simulations.

Hamiltonian Simulation: Trotter-Suzuki Approximations

- Trotter-Suzuki approximations of the evolution operator provide then an efficient quantum algorithm: No complexity is polynomial in the dimension N, only linear in the number of terms in the Hamiltonian K, which scales with the system size n

$$G \sim 5^p K \frac{(Kt)^{1+1/p}}{\epsilon^{1/p}}$$

- No ancillary qubits!

- The method is efficient, but how well does it perform in practice? Polynomial scaling may not be enough if a large number of elementary gates still needs to be implemented to show an advantage with respect to classical-computer simulations.

- A significant body of research still goes to improving methods based on product formulas.

Trotter-Suzuki Approximations in Quantum Chemistry

- Trotter-Suzuki approximations in quantum chemistry (an important problem). This is a fermionic system but can be mapped to a qubit system, as I will explain later. The Hamiltonian is:

$$H = \sum_{i,j=0}^{K-1} c_{ij} \hat{O}_{ij} + \sum_{i,j,k,l=0}^{K-1} c_{ijkl} \hat{O}_{ijkl}$$

Trotter-Suzuki Approximations in Quantum Chemistry

- Trotter-Suzuki approximations in quantum chemistry (an important problem). This is a fermionic system but can be mapped to a qubit system, as I will explain later. The Hamiltonian is:

$$H = \sum_{i,j=0}^{K-1} c_{ij}\hat{O}_{ij} + \sum_{i,j,k,l=0}^{K-1} c_{ijkl}\hat{O}_{ijkl}$$

- Assumptions (for simplicity):

$t := \text{constant}$

$C := \text{constant}$

$\epsilon := \text{constant}$

$\hat{O}_{ij}$, $\hat{O}_{ijkl}$ are Pauli products of at most K qubits

Trotter-Suzuki Approximations in Quantum Chemistry

- Trotter-Suzuki approximations in quantum chemistry (an important problem). This is a fermionic system but can be mapped to a qubit system, as I will explain later. The Hamiltonian is:

$$H = \sum_{i,j=0}^{K-1} c_{ij}\hat{O}_{ij} + \sum_{i,j,k,l=0}^{K-1} c_{ijkl}\hat{O}_{ijkl}$$

- Assumptions (for simplicity):

  $t := \text{constant}$

  $C := \text{constant}$

  $\epsilon := \text{constant}$

  $\hat{O}_{ij}$, $\hat{O}_{ijkl}$ are Pauli products of at most K qubits

- For the 2nd order approximation, where p=2, we obtain $G \propto K^{11}$

Trotter-Suzuki Approximations in Quantum Chemistry

- Trotter-Suzuki approximations in quantum chemistry (an important problem). This is a fermionic system but can be mapped to a qubit system, as I will explain later. The Hamiltonian is:

$$H = \sum_{i,j=0}^{K-1} c_{ij}\hat{O}_{ij} + \sum_{i,j,k,l=0}^{K-1} c_{ijkl}\hat{O}_{ijkl}$$

- To reduce the gate complexity, researchers have used a variety of techniques

2013- Circuit compilation (some gates cancel) $K^{11} \rightarrow K^{10}$

2013/19- Better bounds on nested commutators $K^{10} \rightarrow K^{9}$

2013- Numerical simulations show $K^{9} \rightarrow K^{8}$

Trotter-Suzuki Approximations in Quantum Chemistry

- Trotter-Suzuki approximations in quantum chemistry (an important problem). This is a fermionic system but can be mapped to a qubit system, as I will explain later. The Hamiltonian is:

$$H = \sum_{i,j=0}^{K-1} c_{ij}\hat{O}_{ij} + \sum_{i,j,k,l=0}^{K-1} c_{ijkl}\hat{O}_{ijkl}$$

- To reduce the gate complexity, researchers have used a variety of techniques

2014- Further improvements on the bounds of errors from nested commutators

$$K^8 \rightarrow K^6$$

2014- Improvements in Hamiltonian decompositions

Numerical simulations show better scaling

Trotter-Suzuki Approximations: Some More References

- The original ideas can be found in the papers by Suzuki [1] and some formal results on product formulas that are useful for quantum computing can be found in [2] and other papers

- A careful analysis on errors show that these depend on nested commutators of terms appearing in the Hamiltonian. Many recent improvements exploit the structure of these commutators, either by looking at Lie algebraic properties [3] or other properties like the locality of interactions [4]. A simple randomization method, in which the order of the terms in the Hamiltonian is chosen randomly, improves product formulas a little [5].

[1] M. Suzuki, Phys. Lett. A146, 319 (1990)
[2] D. Berry, G. Ahokas, R. Cleve, and B. Sanders,Comm. Math. Phys.270, 359 (2007)
[3] **R. D. Somma**, J. Math. Phys. 57,062202 (2016)
[4] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, andS. Zhu, arXiv: 1912.08854 (2019)
[5] A. M. Childs, A. Ostrander, Y. Su, Quantum 3, 182 (2019)

Hamiltonian Simulation: Better Approximations?

- Trotter-Suzuki approximations are attractive due to their simplicity
- However, they are not optimal and their scaling with the precision error is poor (polynomial dependence)
- New quantum simulation methods are always useful and can result in novel techniques for quantum computing

Hamiltonian Simulation: Better Approximations?

- Trotter-Suzuki approximations are attractive due to their simplicity
- However, they are not optimal and their scaling with the precision error is poor (polynomial dependence)
- New quantum simulation methods are always useful and can result in novel techniques for quantum computing

- We developed a quantum algorithm that aims at simulating the Taylor series expansion of the evolution operator [6]. Our quantum algorithm is optimal in almost all parameters and has been studied for quantum chemistry with successful results. (Some improvements occurred since then)

[6] D. Berry, A. Childs, R. Cleve, R. Kothari, and **R.D. Somma**, Phys. Rev. Lett. 114, 090502 (2015)

Hamiltonian Simulation: Taylor-series Approximations

- These approximations are also attractive due to their simplicity but they need ancillary qubits and may not be useful in the very near term.
- They basically approximate the evolution operator by implementing the Taylor series expansion of the exponential using 1 and 2-qubit gates.
- An upside is that their complexity in terms of precision is polylogarithmic. This is an exponential improvement over product formulas and this approach will be paramount when highly-precise measurements are needed.
- A downside is that, a priori, they don't exploit the structure of the problem (some current work is going into this). When the structure is unknown, they are provably optimal.

Hamiltonian Simulation: Taylor-series Approximations

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \sum_{r=0}^{R} \frac{(-iHs)^r}{r!} \ , \ s < t \ .$$

Pauli products:
Hermitian and unitary

Hamiltonian Simulation: Taylor-series Approximations

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \; .$$

- $W(s)$ is almost a unitary operation
- $W(s)$ is a linear combination of unitaries after replacing H

Hamiltonian Simulation: Taylor-series Approximations

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \; s < t \; .$$

- $W(s)$ is almost a unitary operation
- $W(s)$ is a linear combination of unitaries after replacing H
- The quantum simulation algorithm aims at implementing the truncated Taylor-series approximation of the exponential operator q times, for $s=t/q$
- It uses two primitives: LCU for linear combination of unitaries and OAA for oblivious amplitude amplification

$$\| U(t) - [W(t/q)]^q \| \leq \epsilon$$

Linear Combination Unitaries (LCU)

- Quantum computers can implement unitary operations. What about linear combinations of unitaries?

Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

$$> 0 \ , \ \alpha + \beta = 1$$

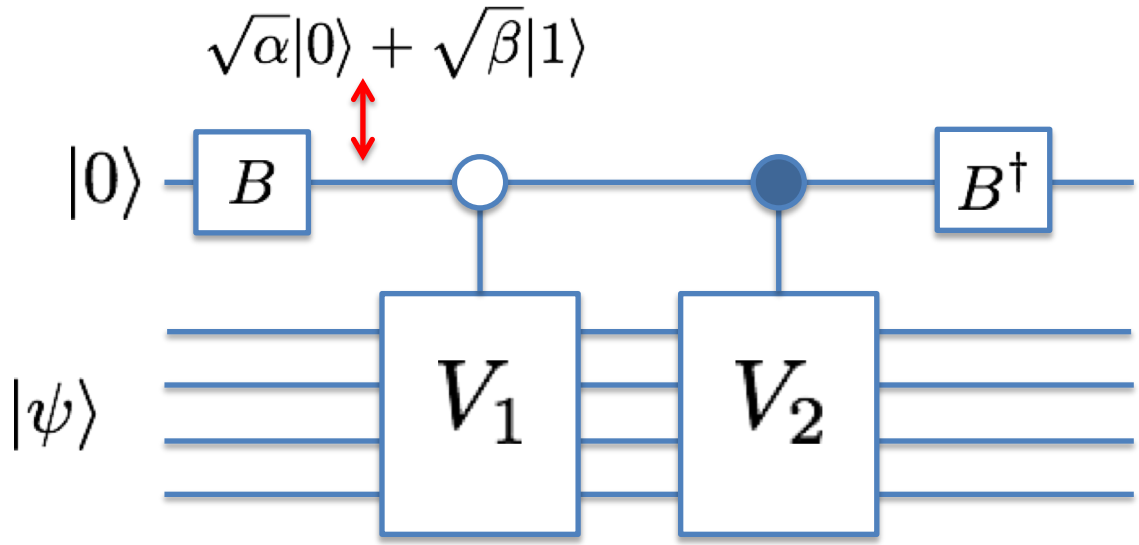
Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

unitaries

Linear Combination Unitaries (LCU)

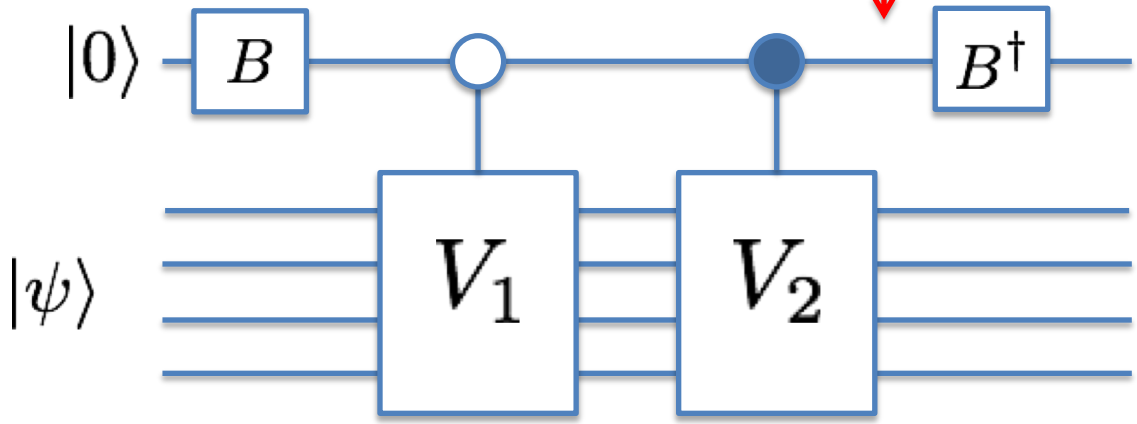
$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$



Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

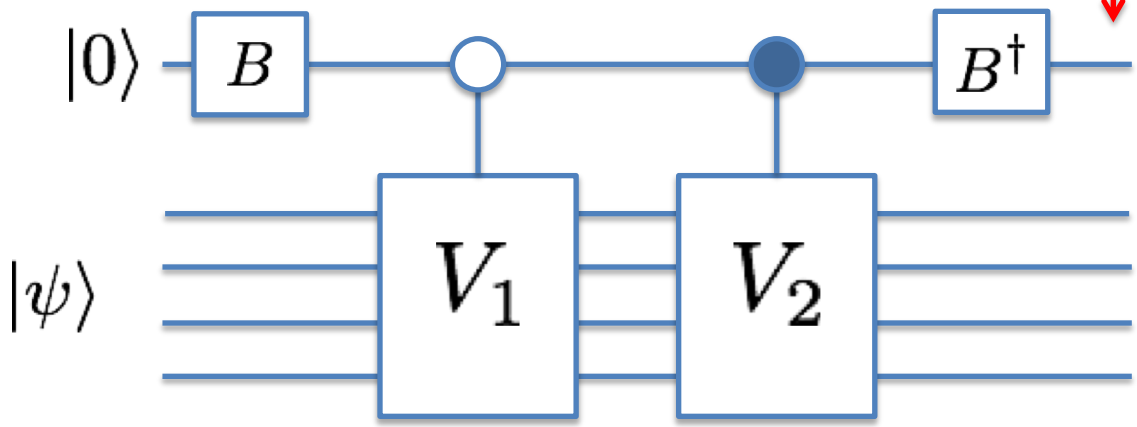
$$\sqrt{\alpha}|0\rangle V_1|\psi\rangle + \sqrt{\beta}|1\rangle V_2|\psi\rangle$$



Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

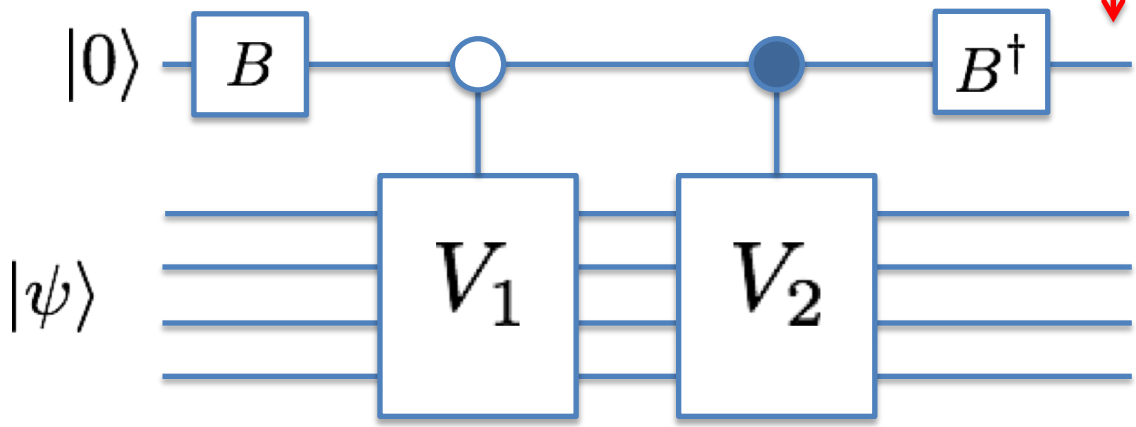
$$|0\rangle(\alpha V_1 + \beta V_2)|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$



Linear Combination Unitaries (LCU)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

$$|0\rangle(\alpha V_1 + \beta V_2)|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$



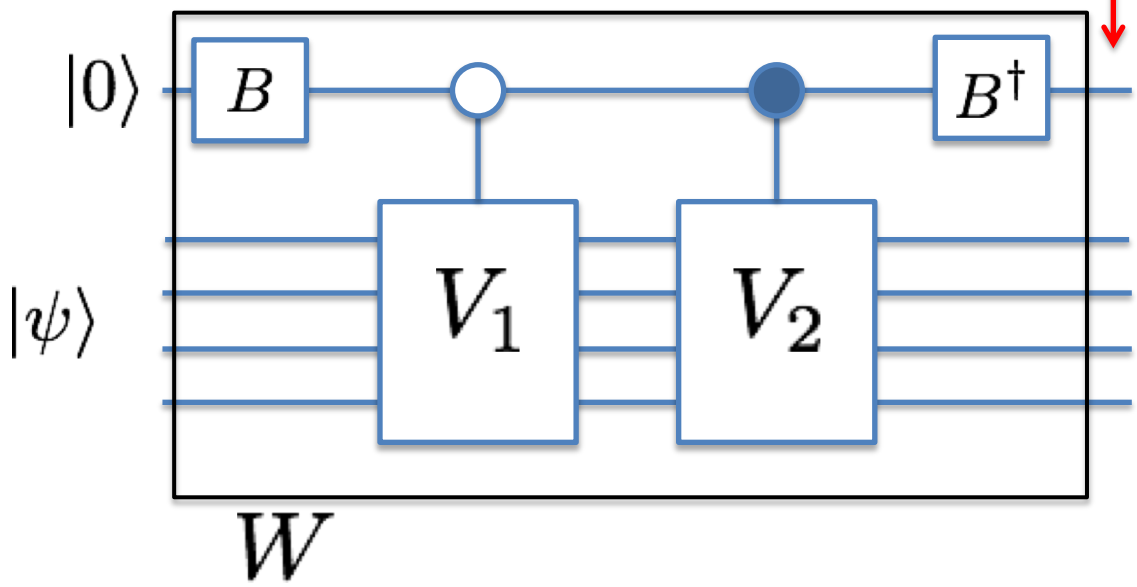
If we measured here, we would obtain the correct state with some probability

$$\Pr(0_{\text{anc}}) = \|(\alpha V_1 + \beta V_2)|\psi\rangle\|^2$$

Oblivious Amplitude Amplification (OAA)

$$|\psi\rangle \to \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

$$|0\rangle(\alpha V_1 + \beta V_2)|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$

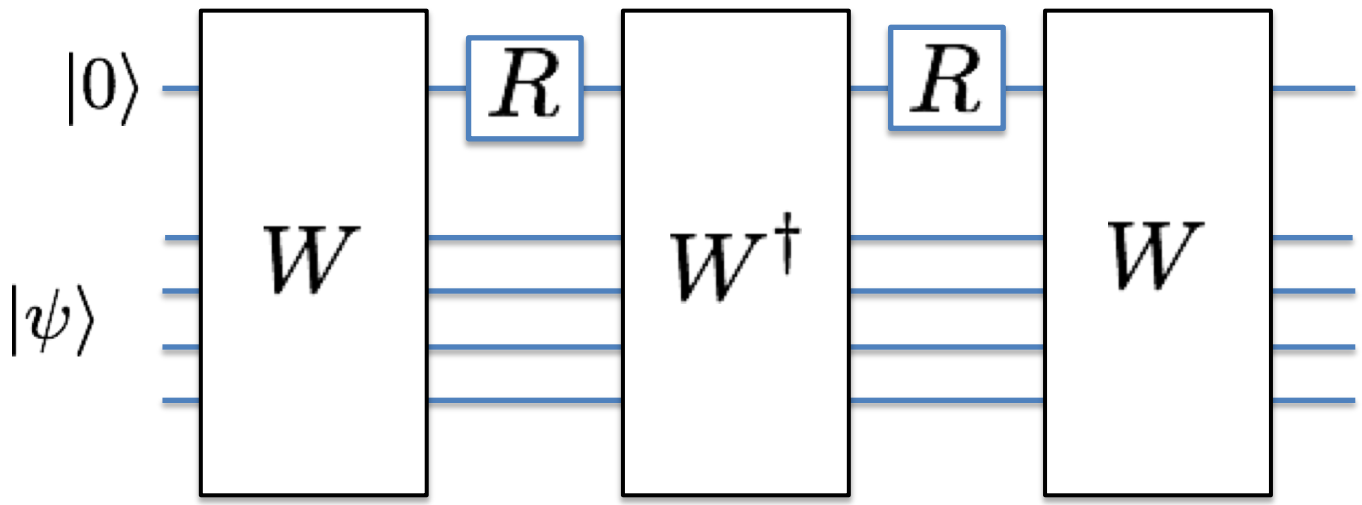


Assume also that $\alpha V_1 + \beta V_2 = \frac{U}{2}$ → unitary $\alpha + \beta = 1$

Oblivious Amplitude Amplification (OAA)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

$|0\rangle U |\psi\rangle$



If we measured here, we would obtain the correct state with probability 1

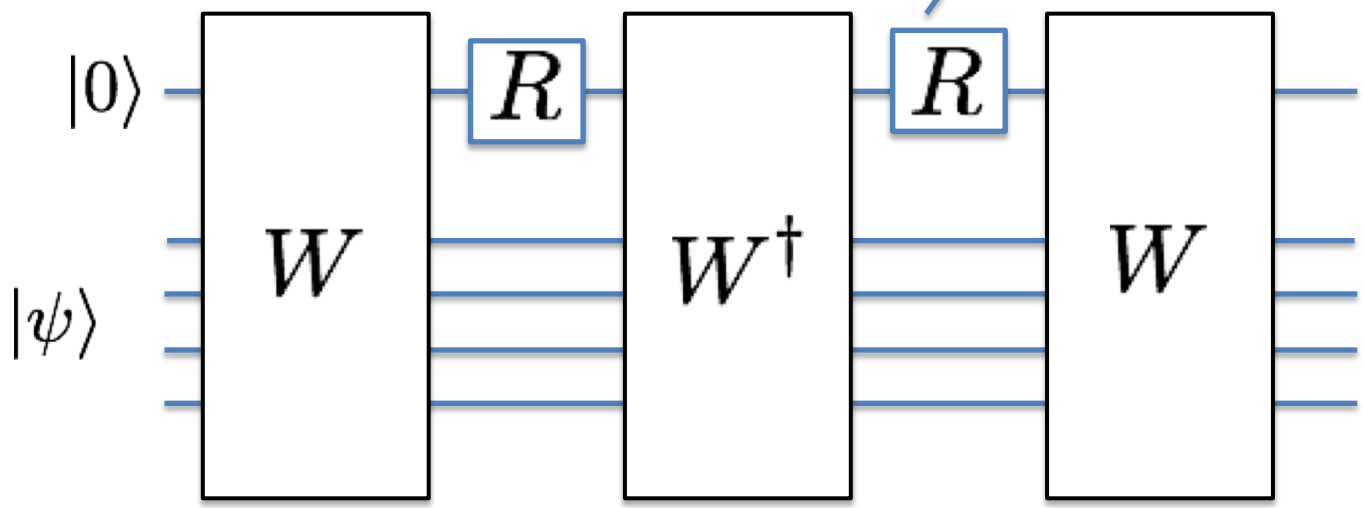
Assume also that $\alpha V_1 + \beta V_2 = \frac{U}{2}$ → unitary $\alpha + \beta = 1$

Oblivious Amplitude Amplification (OAA)

$$|\psi\rangle \rightarrow \frac{(\alpha V_1 + \beta V_2)|\psi\rangle}{\|(\alpha V_1 + \beta V_2)|\psi\rangle\|}$$

The reflection operator:

$$|0\rangle \rightarrow |0\rangle \ , \ |1\rangle \rightarrow -|1\rangle$$



If we measured here, we would obtain the correct state with probability 1

Assume also that $\alpha V_1 + \beta V_2 = \frac{U}{2}$ → unitary $\alpha + \beta = 1$

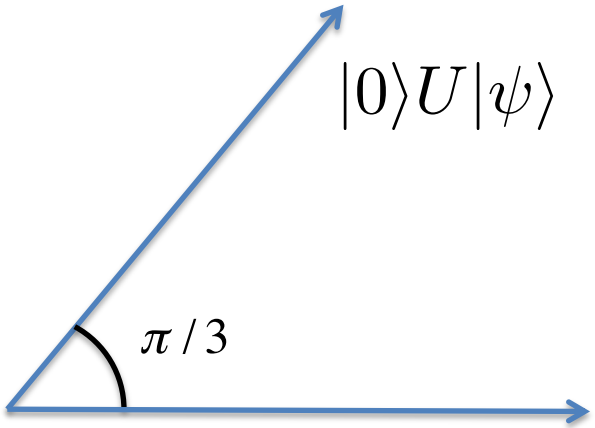
Oblivious Amplitude Amplification (OAA)

The last step was a version of amplitude amplification that works even when the input state is unknown! → “Oblivious amplitude amplification”

$$W|0\rangle|\psi\rangle = |0\rangle(\alpha V_1 + \beta V_2)|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$

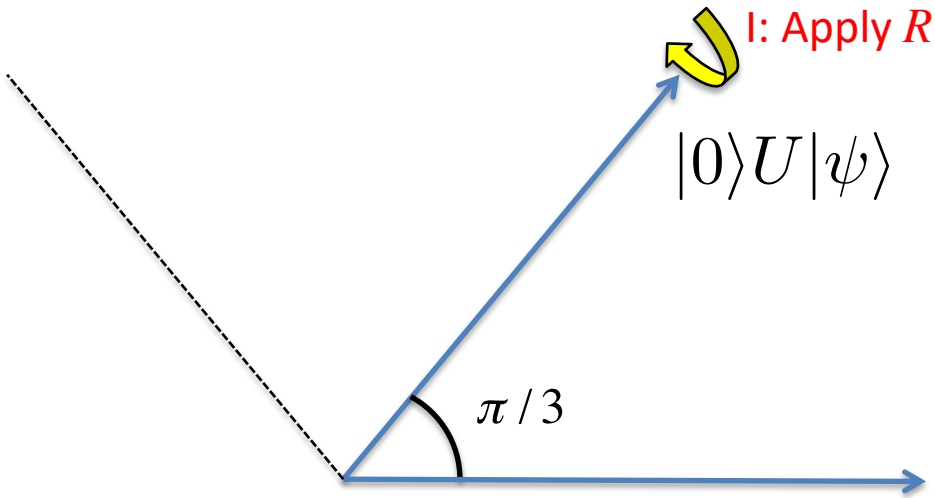
Oblivious Amplitude Amplification (OAA)

$$W|0\rangle|\psi\rangle = |0\rangle(\underbrace{\alpha V_1 + \beta V_2}_{U/2})|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$



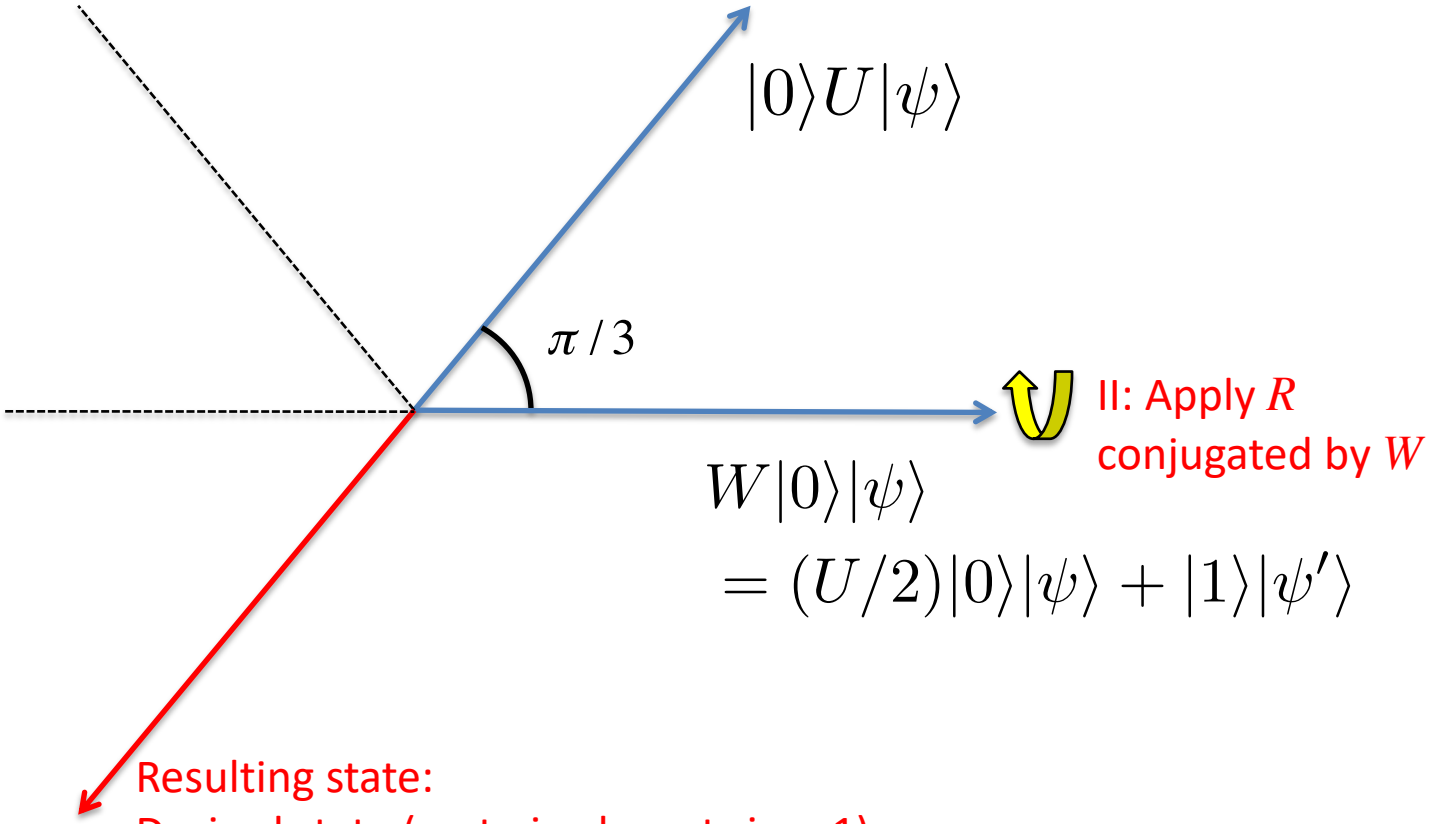
Oblivious Amplitude Amplification (OAA)

$$W|0\rangle|\psi\rangle = |0\rangle\underbrace{(\alpha V_1 + \beta V_2)}_{U/2}|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$



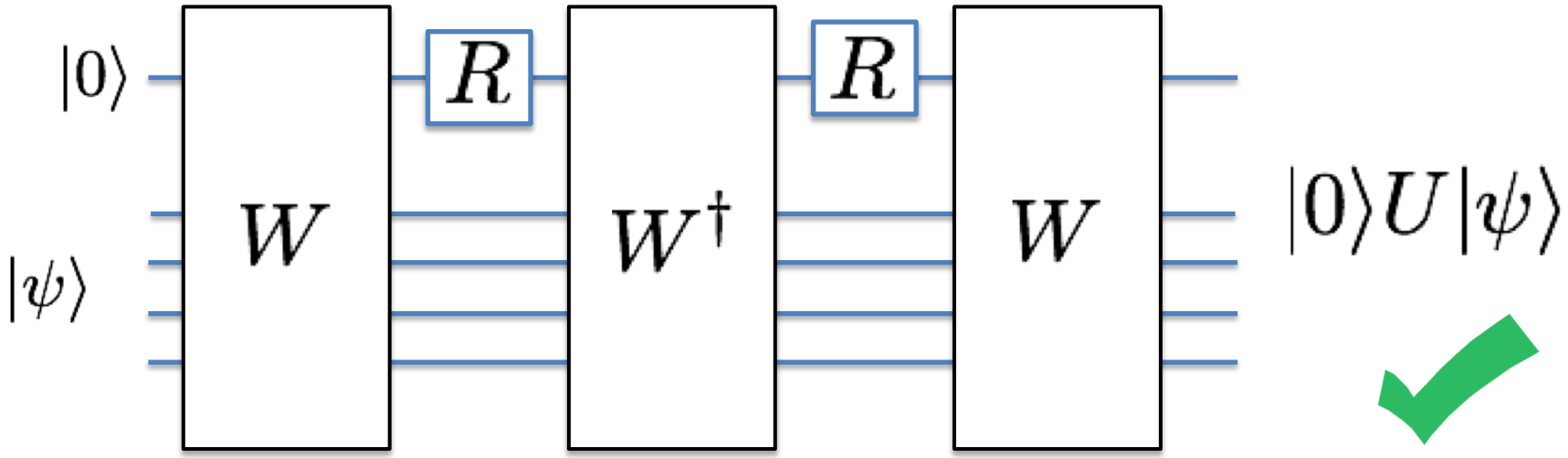
Oblivious Amplitude Amplification (OAA)

$$W|0\rangle|\psi\rangle = |0\rangle\underbrace{(\alpha V_1 + \beta V_2)}_{U/2}|\psi\rangle + \sqrt{\alpha\beta}|1\rangle(V_1 - V_2)|\psi\rangle$$



Oblivious Amplitude Amplification (OAA)

$$|\psi\rangle \rightarrow U|\psi\rangle$$



Assume also that $\alpha V_1 + \beta V_2 = \frac{U}{2}$ → unitary $\alpha + \beta = 1$

LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

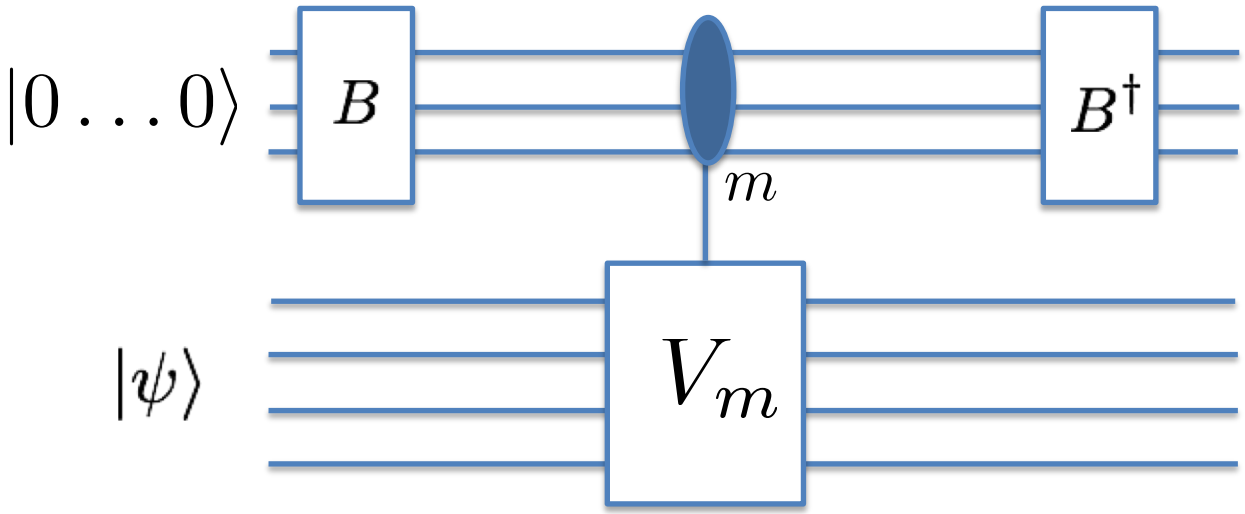
Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$

LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$

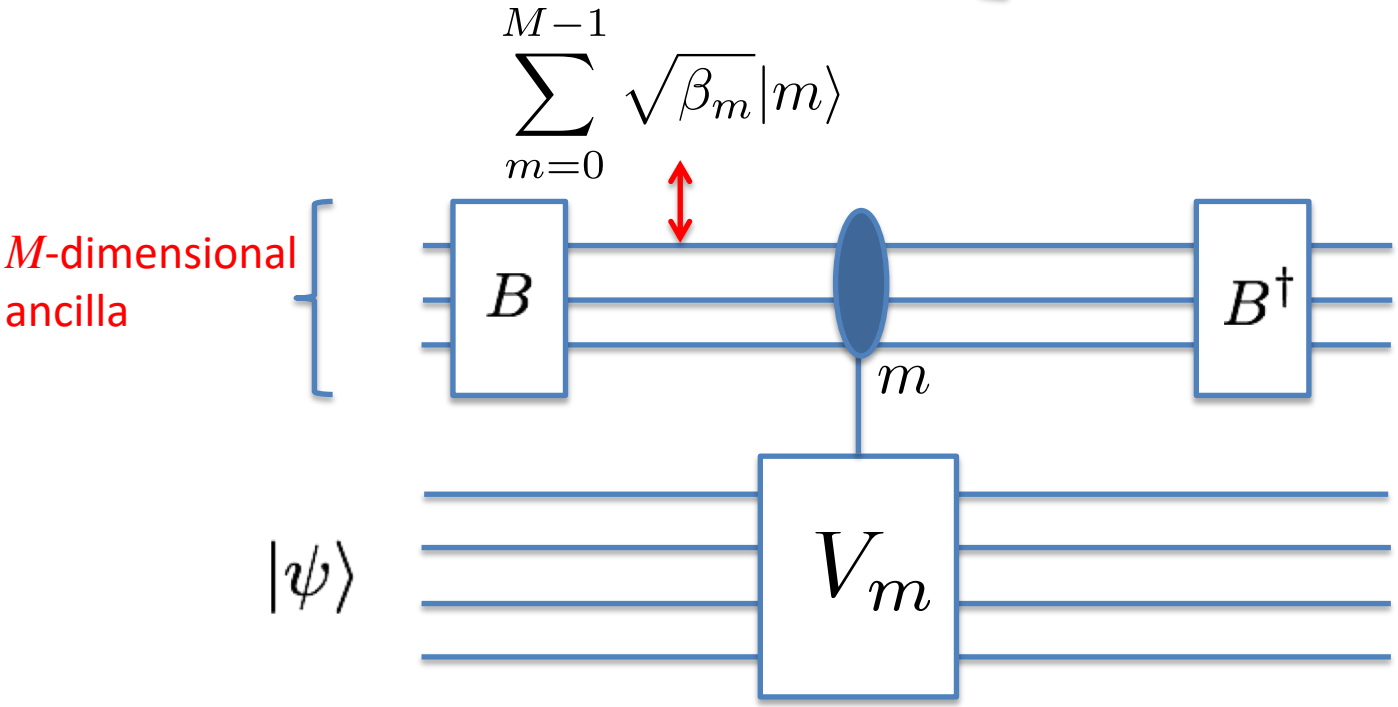


LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$



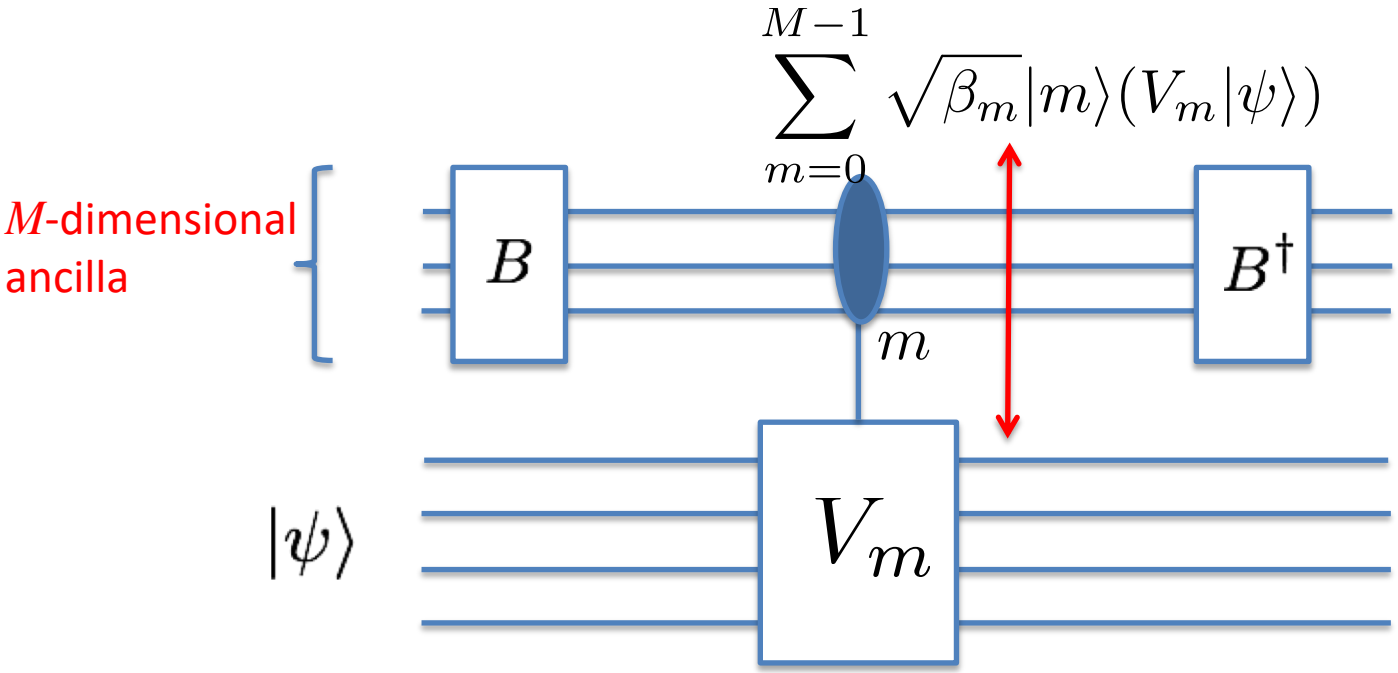
LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary

$$\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$$

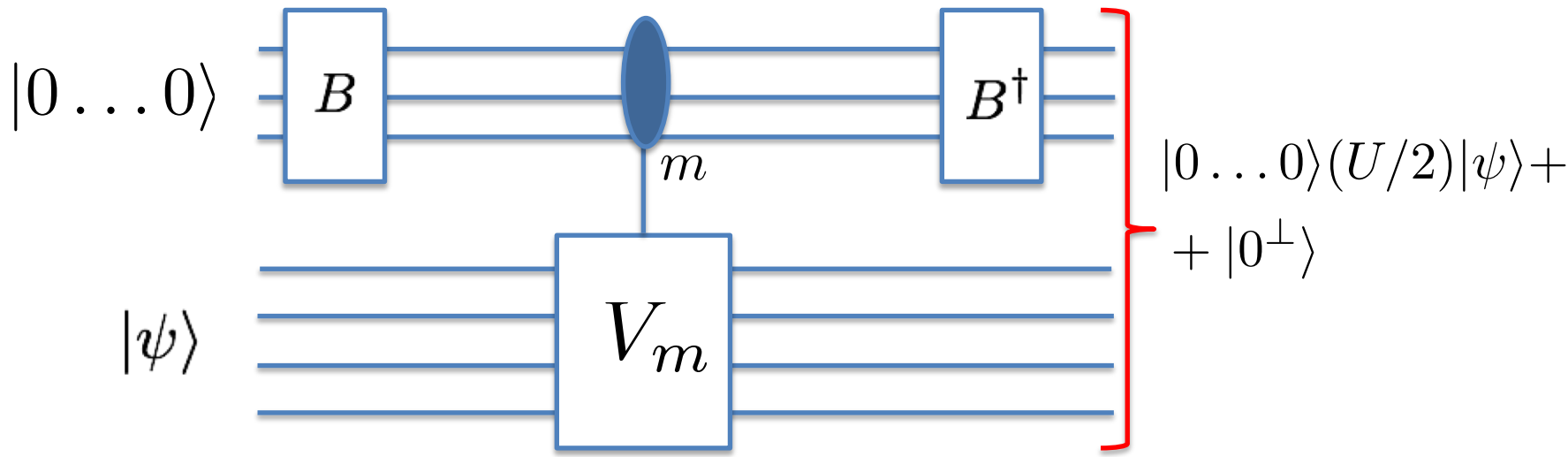


LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$

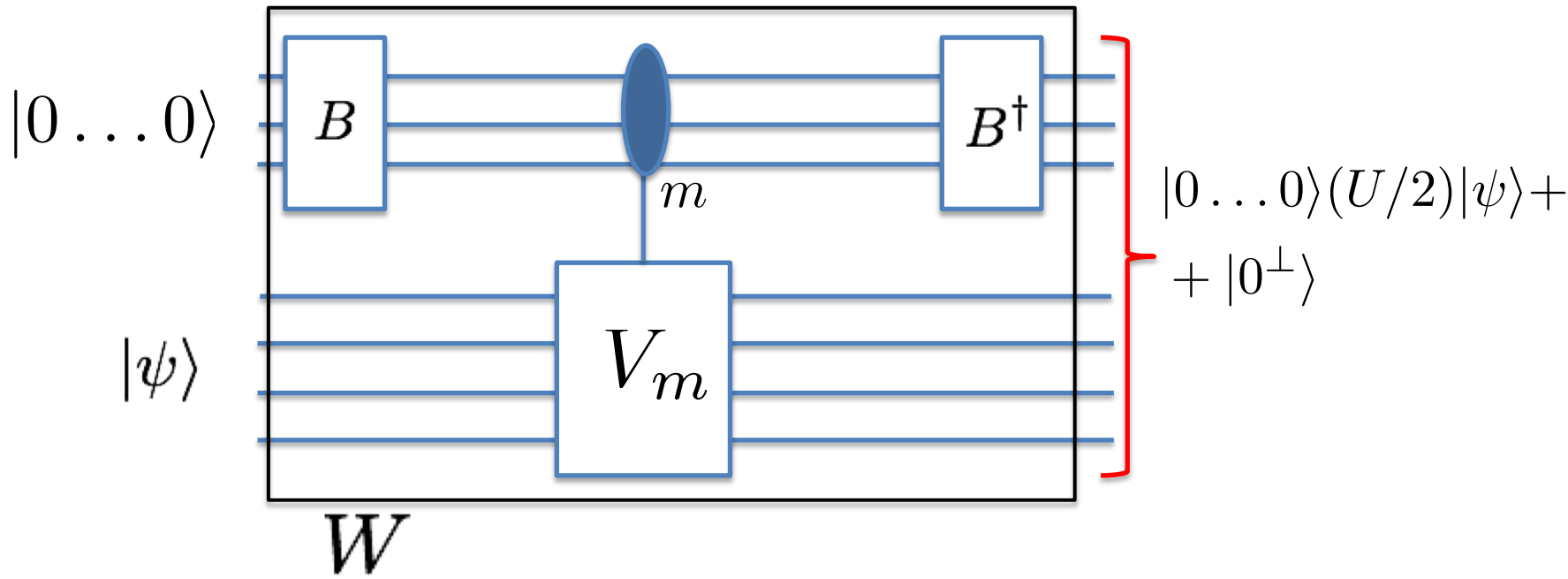


LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$



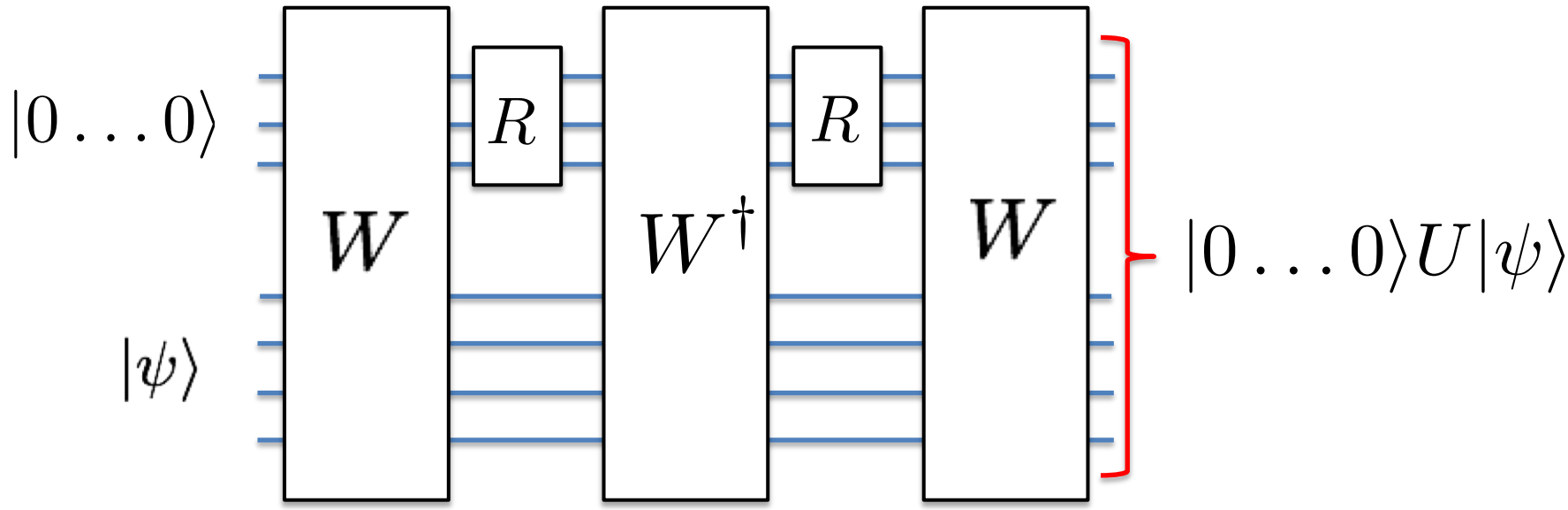
LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$

Oblivious amplitude amplification



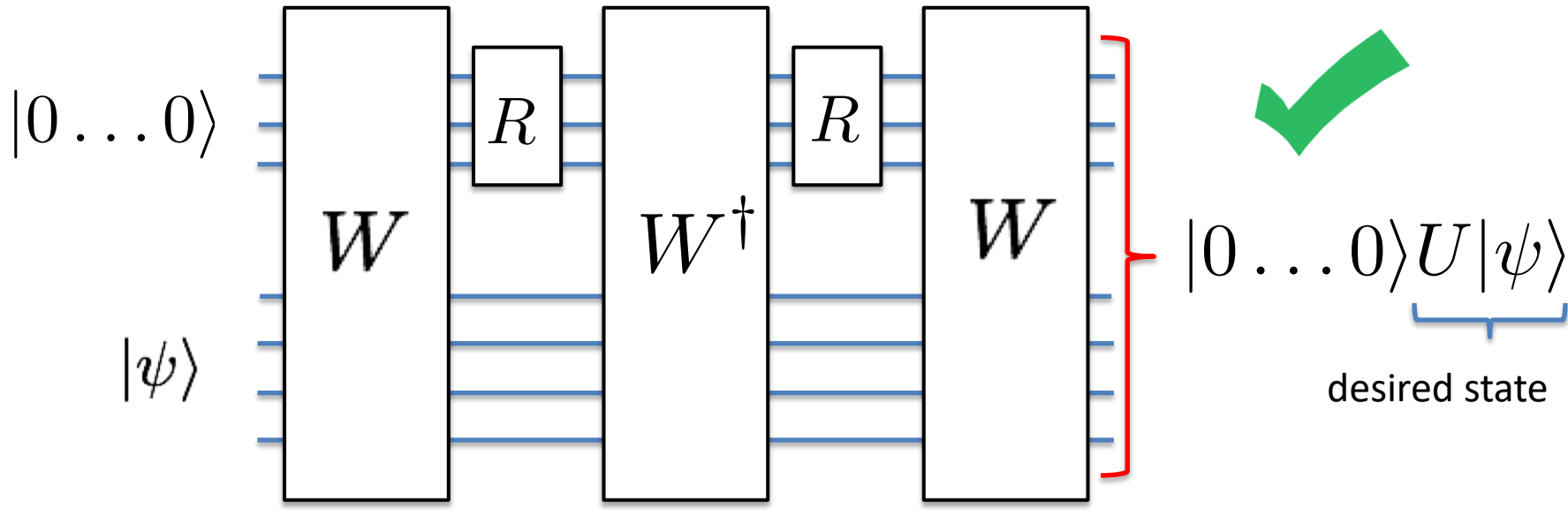
LCU and OAA

In a more general case:

$$\frac{U}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Half a unitary $\begin{cases} \beta_m > 0 \\ \sum_{m=0}^{M-1} \beta_m = 1 \end{cases}$

Oblivious amplitude amplification



Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \; s < t \; .$$

Pauli products:
Hermitian and unitary

$$|c_k| \leq 1$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \, .$$

Pauli products:
Hermitian and unitary
$|c_k| \leq 1$

- First we obtain the LCU:

$$W(s) = \sum_{r=0}^{R} \sum_{k_1,\ldots,k_r=0}^{K-1} \frac{(-is)^r}{r!} c_{k_1} \ldots c_{k_r} \hat{O}_{k_1} \ldots \hat{O}_{k_r}$$

$$= \sum_{m=0}^{M-1} \beta'_m V_m$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \, .$$

Pauli products:
Hermitian and unitary
$|c_k| \leq 1$

- First we obtain the LCU:

$$W(s) = \sum_{r=0}^{R} \sum_{k_1,\ldots,k_r=0}^{K-1} \frac{(-is)^r}{r!} c_{k_1} \ldots c_{k_r} \hat{O}_{k_1} \ldots \hat{O}_{k_r}$$

$$= \sum_{m=0}^{M-1} \beta'_m V_m$$

β'_m: >0

V_m: Unitaries: products of Pauli operators

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \, .$$

- First we obtain the LCU:

$$W(s) = \sum_{r=0}^{R} \sum_{k_1,\ldots,k_r=0}^{K-1} \frac{(-is)^r}{r!} c_{k_1} \ldots c_{k_r} \hat{O}_{k_1} \ldots \hat{O}_{k_r}$$

$$= \sum_{m=0}^{M-1} \beta'_m V_m$$

$$M = 1 + K + \ldots + K^R$$

$$\text{Picking } s \approx \ln 2 / \sum_k c_k \quad \Rightarrow \quad \sum_m \beta'_m \approx 2$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \; .$$

- First we obtain the LCU: $s \approx \ln 2 / \sum_k c_k$

Then if $\beta_m = \beta'_m/2$, we obtain $\sum_m \beta_m \approx 1, \; \beta_m > 0$

and $$\frac{W(s)}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \, .$$

- First we obtain the LCU: $s \approx \ln 2 / \sum_k c_k$

Then if $\beta_m = \beta'_m/2$, we obtain $\sum_m \beta_m \approx 1, \; \beta_m > 0$

and $\dfrac{W(s)}{2} = \sum_{m=0}^{M-1} \beta_m V_m$

Since this is a good approximation to the evolution operator $U(t)$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \, .$$

- First we obtain the LCU: $s \approx \ln 2 / \sum_k c_k$

$$\text{Then if } \beta_m = \beta'_m/2, \text{ we obtain } \sum_m \beta_m \approx 1, \; \beta_m > 0$$

$$\text{and } \frac{W(s)}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

Since this is a good approximation to the evolution operator $U(t)$

Then this LCU is almost half a unitary

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)}, \; s < t \; .$$

- First we obtain the LCU: $s \approx \ln 2 / \sum_k c_k$

Then if $\beta_m = \beta'_m/2$, we obtain $\sum_m \beta_m \approx 1, \; \beta_m > 0$

and $$\frac{W(s)}{2} = \sum_{m=0}^{M-1} \beta_m V_m$$

We picked s to fit the LCU and OAA approaches

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \; s < t \; .$$

- Second, for that choice of s we note that the number of steps is $q=t/s$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \underbrace{\hat{O}_k}_{\text{Pauli products: Hermitian and unitary}} \quad \Rightarrow \quad U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \; s < t \; .$$

- Second, for that choice of s we note that the number of steps is $q=t/s$

$$\Rightarrow \quad q = \mathcal{O}(tK)$$

(Because *s ~ 1/K)*

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

Pauli products:
Hermitian and unitary

$$U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \quad s < t \, .$$

- Second, for that choice of s we note that the number of steps is $q=t/s$

$$\Rightarrow \quad q = \mathcal{O}(tK)$$

- Then we need to pick the approximation order R such that

$$\|W(s) - U(s)\| \leq \epsilon/q$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

Pauli products:
Hermitian and unitary

$$U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \quad s < t .$$

- Second, for that choice of s we note that the number of steps is $q=t/s$

$$\Rightarrow \quad q = \mathcal{O}(tK)$$

- Then we need to pick the approximation order R such that

$$\|W(s) - U(s)\| \leq \epsilon/q$$

$$\Rightarrow \quad \boxed{R = O\left(\frac{\log(q/\epsilon)}{\log\log(q/\epsilon)}\right)}$$

Hamiltonian Simulation: Taylor series, LCU, and OAA

We were here:

$$H = \sum_{k=0}^{K-1} c_k \hat{O}_k$$

Pauli products:
Hermitian and unitary

$$U(s) = e^{-iHs} \approx \underbrace{\sum_{r=0}^{R} \frac{(-iHs)^r}{r!}}_{W(s)} , \quad s < t .$$

- Second, for that choice of s we note that the number of steps is $q=t/s$

$$\Rightarrow \quad q = \mathcal{O}(tK)$$

- Then we need to pick the approximation order R such that

$$\|W(s) - U(s)\| \leq \epsilon/q$$

$$\Rightarrow \quad R = \mathcal{O}\left(\frac{\log(tK/\epsilon}{\log\log(tK/\epsilon)}\right)$$

Hamiltonian Simulation: Taylor series algorithm

$$|\psi\rangle \rightarrow\approx [W(t/q)]^q |\psi\rangle \approx U(t)|\psi\rangle$$

- The gate complexity of the LCU + OAA method is

  G= Number of steps (q) **x** complexity of each step

  Complexity of each step = 3 **x** cutoff order **x** complexity implementing LCU for H

Hamiltonian Simulation: Taylor series algorithm

$$|\psi\rangle \rightarrow \approx [W(t/q)]^q |\psi\rangle \approx U(t)|\psi\rangle$$

- The gate complexity of the LCU + OAA method is

  G= Number of steps (q) **x** complexity of each step

  Complexity of each step = 3 **x** cutoff order **x** complexity implementing LCU for H

$$G = \mathcal{O}\left(tK^2 \frac{\log(tK/\epsilon)}{\log\log(tK/\epsilon)} n \right)$$

Hamiltonian Simulation: Taylor series algorithm and Optimality

$$G = \mathcal{O}\left(tK^2 \frac{\log(tK/\epsilon)}{\log\log(tK/\epsilon)} n\right)$$

- Strict improvement over previously known approaches for Hamiltonian simulation
- The gate complexity G depends logarithmically in $1/\varepsilon$ and is almost linear in t
- The gate complexity G is optimal: There is a matching lower bound
- LCU and OAA are new techniques that encountered applications in several other quantum algorithms (Gibbs state preparation, solving linear systems,…)

Hamiltonian Simulation: Taylor series algorithm for Quantum Chemistry

2015- Further improvements on this problem were obtained using the described simulation algorithm

$$K^6 \rightarrow K^5$$

Exponentially more precise quantum simulation of fermions I: Quantum chemistry in second quantization

Ryan Babbush,[1,2,*] Dominic W. Berry,[3,†] Ian D. Kivlichan,[1,4]
Annie Y. Wei,[1] Peter J. Love,[5] and Alán Aspuru-Guzik[1,‡]

[1] *Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138*
[2] *Google, Venice, CA 90291, USA*
[3] *Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia*
[4] *Department of Physics, Harvard University, Cambridge, MA 02138, USA*
[5] *Department of Physics and Astronomy, Tufts University, Medford, MA 02155*

(Dated: September 30, 2015)

We introduce novel algorithms for the quantum simulation of molecular systems which are asymptotically more efficient than those based on the Trotter-Suzuki decomposition. We present the first application of a recently developed technique for simulating Hamiltonian evolution using a truncated Taylor series to obtain logarithmic scaling with the inverse of the desired precision, an exponential improvement over all prior methods. The two algorithms developed in this work rely on a second quantized encoding of the wavefunction in which the state of an N spin-orbital system is encoded in $\mathcal{O}(N)$ qubits. Our first algorithm requires at most $\widetilde{\mathcal{O}}(N^8 t)$ gates. Our second algorithm involves on-the-fly computation of molecular integrals, in a way that is exponentially more precise than classical sampling methods, by using the truncated Taylor series simulation technique. Our second algorithm has the lowest gate count of any approach to second quantized quantum chemistry simulation in the literature, scaling as $\widetilde{\mathcal{O}}(N^5 t)$. The approaches presented here are readily applicable to a wide class of fermionic models, many of which are defined by simplified versions of the chemistry Hamiltonian.

Hamiltonian Simulation: Improvements on the Taylor series algorithm?

PRL 118, 010501 (2017) PHYSICAL REVIEW LETTERS week ending 6 JANUARY 2017

Optimal Hamiltonian Simulation by Quantum Signal Processing

Guang Hao Low and Isaac L. Chuang
Department of Physics, Center for Ultracold Atoms, and Research Laboratory of Electronics Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

(Received 8 June 2016; revised manuscript received 13 November 2016; published 5 January 2017)

The physics of quantum mechanics is the inspiration for, and underlies, quantum computation. As such, one expects physical intuition to be highly influential in the understanding and design of many quantum algorithms, particularly simulation of physical systems. Surprisingly, this has been challenging, with current Hamiltonian simulation algorithms remaining abstract and often the result of sophisticated but unintuitive constructions. We contend that physical intuition can lead to optimal simulation methods by showing that a focus on simple single-qubit rotations elegantly furnishes an optimal algorithm for Hamiltonian simulation, a universal problem that encapsulates all the power of quantum computation. Specifically, we show that the query complexity of implementing time evolution by a d-sparse Hamiltonian $\hat{H}$ for time-interval t with error ϵ is $\mathcal{O}[td\|\hat{H}\|_{\max} + \log(1/\epsilon)/\log\log(1/\epsilon)]$, which matches lower bounds in all parameters. This connection is made through general three-step "quantum signal processing" methodology, comprised of (i) transducing eigenvalues of $\hat{H}$ into a single ancilla qubit, (ii) transforming these eigenvalues through an optimal-length sequence of single-qubit rotations, and (iii) projecting this ancilla with near unity success probability.

DOI: 10.1103/PhysRevLett.118.010501

Less ancillary qubits suffice

Hamiltonian Simulation via Quantum Signal Processing

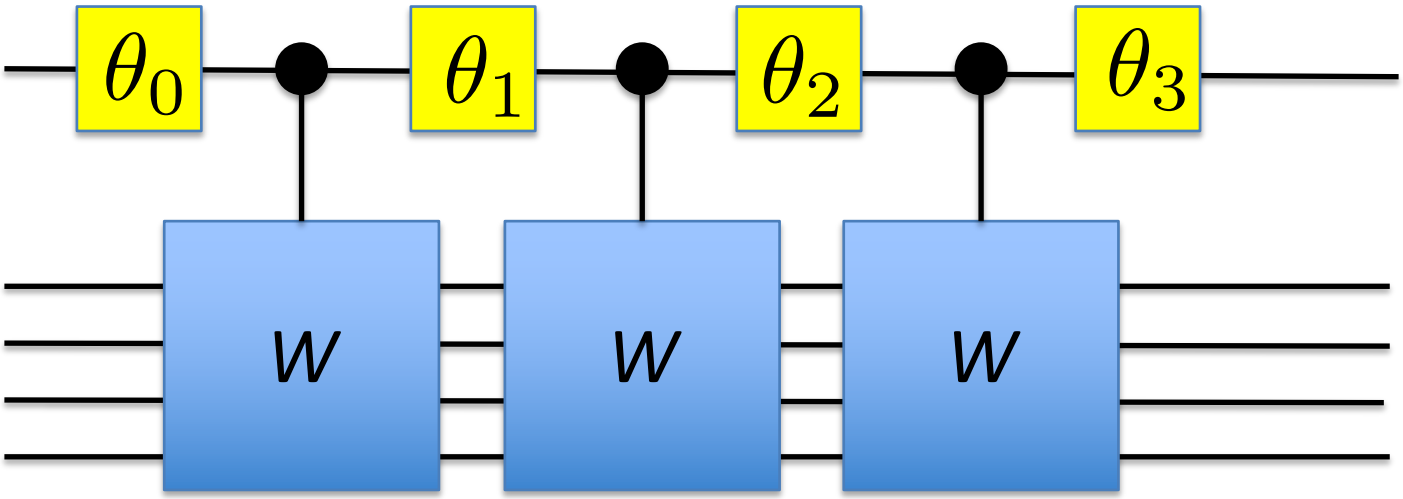
Rough idea:

$$|0..0\rangle U(t)|\psi\rangle \approx \sum_j \alpha_j W^j |0..0\rangle|\psi\rangle$$

Hamiltonian Simulation via Quantum Signal Processing

Rough idea: $$|0..0\rangle U(t)|\psi\rangle \approx \sum_j \underbrace{\alpha_j W^j}|0..0\rangle|\psi\rangle$$

Polynomial that depends on *H*



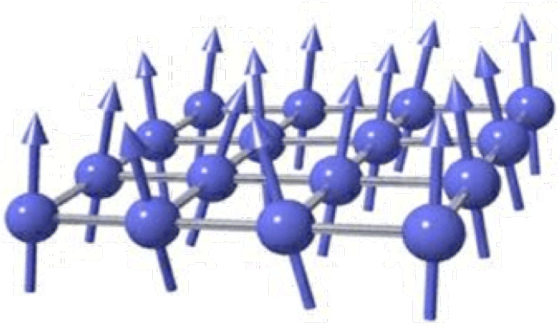
Approximates the polynomial

• • •

Problem: Given the coefficients in the polynomial, find the single qubit rotations
This may be done efficiently; recent progress by Microsoft and Alibaba

Logarithmic scaling in precision due to efficient polynomial approximations

Quantum Simulations: What other methods exist?



- The evolution operator can also be simulated by using a quantum walk operator combined with quantum phase estimation. This approach results in a nice scaling in terms of t (almost linear) but not in terms of ε [Berry,Childs (2012)].

- For some specific Hamiltonians, other and more efficient approaches may work. Gate optimization techniques are also important, and a lot of recent work has gone into optimizing quantum circuits.

Hamiltonian Simulation: The Query Model

- So far we assumed that the Hamiltonian was presented as $H = \sum_{k=0}^{K-1} c_k \hat{O}_k$

- The query model does not make such an assumption. Instead, it assumes that we can query the matrix elements of *H* – a matrix of exponential dimension

Hamiltonian Simulation: The Query Model

- So far we assumed that the Hamiltonian was presented as $H = \sum_{k=0}^{K-1} c_k \hat{O}_k$

- The query model does not make such an assumption. Instead, it assumes that we can query the matrix elements of *H* – a matrix of exponential dimension

- The assumption is that we have access to an oracle O_H such that

$$O_H|j,k\rangle = |j, v(j,k)\rangle$$

Hamiltonian Simulation: The Query Model

- So far we assumed that the Hamiltonian was presented as $H = \sum_{k=0}^{K-1} c_k \hat{O}_k$

- The query model does not make such an assumption. Instead, it assumes that we can query the matrix elements of *H* – a matrix of exponential dimension

- The assumption is that we have access to an oracle O_H such that

$$O_H|j,k\rangle = |j, v(j,k)\rangle$$

row index

element index

column index

Hamiltonian Simulation: The Query Model

- So far we assumed that the Hamiltonian was presented as $H = \sum_{k=0}^{K-1} c_k \hat{O}_k$

- The query model does not make such an assumption. Instead, it assumes that we can query the matrix elements of *H* – a matrix of exponential dimension

- The assumption is that we have access to an oracle O_H such that

$$O_H|j,k\rangle = |j, v(j,k)\rangle$$

row index — element index — column index

$$O_H|j,j',z\rangle = |j,j',H_{jj'} \oplus z\rangle$$

row / column index — q-bit string — matrix element

Hamiltonian Simulation: The Query Model

- We can use the previous results to provide a quantum algorithm that approximates time evolution
- The quantum algorithm uses the oracle and other 2-qubit gates
- The complexity is determined by two components: The query complexity, which is the number of calls to the oracle, and the gate complexity, which is the additional number of 2-qubit gates that are independent of the Hamiltonian
- There are various ways of constructing the algorithm. A common one is to consider the adjacency graph and split H into a combination of Hamiltonians, each acting on 2-dimensional subspaces. This is done via a particular coloring.

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent

- However, methods based on Trotter-Suzuki approximations and Taylor series approximations can be generalized to the time-dependent case

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, methods based on Trotter-Suzuki approximations and Taylor series approximations can be generalized to the time-dependent case
- Trotter-Suzuki methods will have an increase in the gate complexity that is polynomial in the norm of the rate of change of the Hamiltonian, which is bad

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, methods based on Trotter-Suzuki approximations and Taylor series approximations can be generalized to the time-dependent case
- Trotter-Suzuki methods will have an increase in the gate complexity that is polynomial in the norm of the rate of change of the Hamiltonian, which is bad
- Methods based on the Taylor series have a small increase in complexity that is only logarithmic in the norm of the rate of change of the Hamiltonian

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, methods based on Trotter-Suzuki approximations and Taylor series approximations can be generalized to the time-dependent case
- Trotter-Suzuki methods will have an increase in the gate complexity that is polynomial in the norm of the rate of change of the Hamiltonian, which is bad
- Methods based on the Taylor series have a small increase in complexity that is only logarithmic in the norm of the rate of change of the Hamiltonian
- To achieve this nice scaling, rather than approximating the Taylor series of the exponential we need to approximate the time-ordered series of the evolution operator as a linear combination of unitaries

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, the presented simulation methods can also be used for fermionic, anyonic, and even bosonic systems [Somma++ (2002)]

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, the presented simulation methods can also be used for fermionic, anyonic, and even bosonic systems [Somma++ (2002)]
- For fermionic and anyonic systems, we can use the so-called Jordan-Wigner mappings that map the fermionic or anyonic algebra to the Pauli algebra

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, the presented simulation methods can also be used for fermionic, anyonic, and even bosonic systems [Somma++ (2002)]
- For fermionic and anyonic systems, we can use the so-called Jordan-Wigner mappings that map the fermionic or anyonic algebra to the Pauli algebra
- In this way, we reduce the problem to that of simulating qubits

Hamiltonian Simulation: What I didn't talk about…

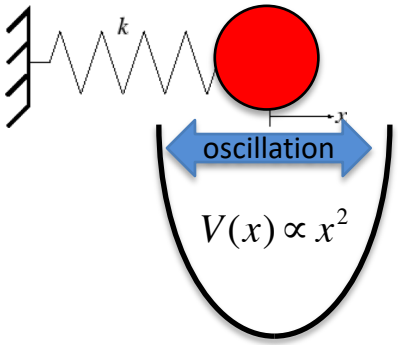
- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- However, the presented simulation methods can also be used for fermionic, anyonic, and even bosonic systems [Somma++ (2002)]
- For fermionic and anyonic systems, we can use the so-called Jordan-Wigner mappings that map the fermionic or anyonic algebra to the Pauli algebra
- In this way, we reduce the problem to that of simulating qubits
- Simulating bosons is more complicated and we need to set bounds on the number of bosons in the system since we are working with finite degrees of freedom, but still can be done

Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent

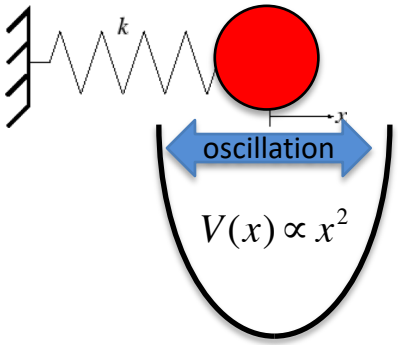
Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- The presented simulation methods may also be used to simulate quantum systems in continuous variables, such as a particle in a potential [Somma(2015)]



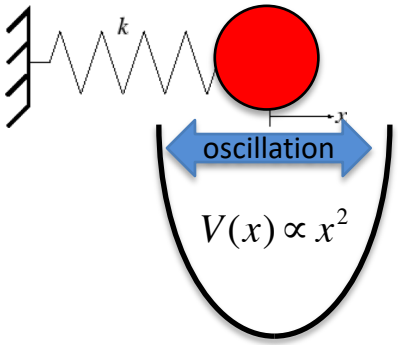
Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- The presented simulation methods may also be used to simulate quantum systems in continuous variables, such as a particle in a potential [Somma(2015)]
- A lot of recent work has gone to Hamiltonian simulation methods for quantum field theories and other CV systems



Hamiltonian Simulation: What I didn't talk about…

- We assumed that the Hamiltonian was for a qubits' system and was time-independent
- The presented simulation methods may also be used to simulate quantum systems in continuous variables, such as a particle in a potential [Somma(2015)]
- A lot of recent work has gone to Hamiltonian simulation methods for quantum field theories and other CV systems
- Some complications come from the fact that we need to discretize the system and work with finite degrees of freedom



Thank you

Los Alamos NATIONAL LABORATORY EST. 1943

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

U.S. DEPARTMENT OF ENERGY | Office of Science

Hamiltonian Simulation: Some Useful References

QS by Trotter-Suzuki Approximations:

- D. Berry, G. Ahokas, R. Cleve, B. Sanders, CMP 270, 359 (2007)
- N. Wiebe, D. Berry, P. Hoyer, B. Sanders, J. Phys. A 43, 065203 (2010)

QS by the Taylor series method:

- D. Berry, A. Childs, R. Cleve, R. Kothari, and **RDS**, Phys. Rev. Lett. 114, 090502 (2015)
- D. Berry, A. Childs, R. Cleve, R. Kothari, and **RDS**, STOC 2014, pp 283-292 (2014)

Quantum algorithms for quantum chemistry:

- M. B. Hastings, D. Wecker, B. Bauer and M. Troyer, Quant. Inf. Comp. 15, 1 (2015)
- R. Babbush, et.al., New J. Phys. 18, 033032 (2016)

Quantum algorithms for fermionic and bosonic systems:

- **RDS**, G,. Ortiz, E. Knill, and J. Gubernatis, Int. J. Quant. Inf. 1, 189 (2003)

QS based on quantum walks:

- D. Berry and A. Childs, Quant. Inf. Comp. 12, 29 (2012)

Quantum algorithms to prepare thermal states:

- A. Chowdhury and **RDS**, Quant. Inf. Comp. 17, 0041 (2017)
- **RDS**, S. Boixo, H. Barnum, and E. Knill, Phys. Rev. Lett. 101, 130504 (2008).
- D. Poulin and P. Wocjan, Phys. Rev. Lett. 103, 220502 (2009).